EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

S1D13708
TECHNICAL MANUAL

Document Number: X39A-Q-001-01

THIS PAGE LEFT BLANK

COMPREHENSIVE SUPPORT TOOLS

EPSON provides the designer and manufacturer a complete set of resources and tools for the development of LCD Graphics Systems.

Documentation

- Technical manuals
- Evaluation/Demonstration board manual

Evaluation/Demonstration Board

- Assembled and fully tested Graphics Evaluation/Demonstration board
- Schematic of Evaluation/Demonstration board
- Parts List
- Installation Guide
- CPU Independent Software Utilities
- Evaluation Software
- Display Drivers

Application Engineering Support

EPSON offers the following services through their Sales and Marketing Network:

- Sales Technical Support
- Customer Training
- Design Assistance

Application Engineering Support

Engineering and Sales Support is provided by:

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com/

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

THIS PAGE LEFT BLANK

S1D13708 Embedded Memory LCD Controller

July 2001

The S1D13708 is a color/monochrome LCD graphics controller with an embedded memory / display buffer. Targeted at PDA and Cell Phone applications, the S1D13708 'directly' interfaces to numerous TFT panels and incorporates a minimum pin-count CPU interface thereby making it an ideal solution for an LCD Module.

This high level of integration combined with a 1.8V Core, provides a low cost, low power, single chip solution to meet the demands of embedded markets such as Mobile Communications devices and Palm-size PCs, where board size and battery life are major concerns.

The embedded display buffer greatly improves overall system performance as the S1D13708 handles all of the display functions directly with very little interaction from the processor.

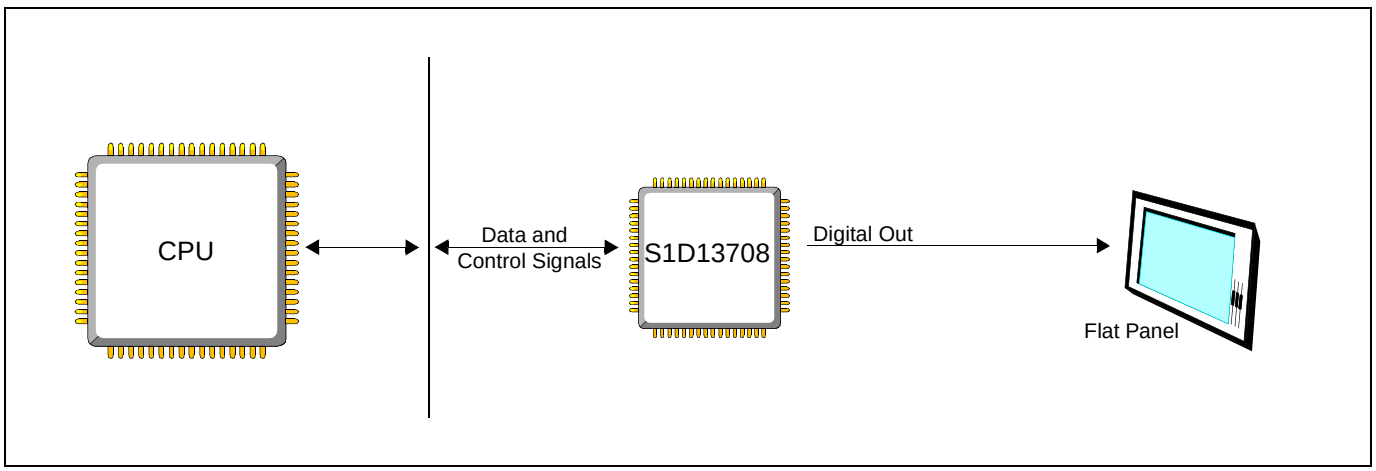
The S1D13708 provides very flexible display features, from our patented SwivelView™ technology which provides hardware rotation of the displayed image, to our Ink Layer with transparency, to our "Picture-in-Picture Plus" feature which allows two active variable size display 'windows'.

The S1D13708 provides impressive support for Mobile Communication devices and Palm OS® handhelds, however its impartiality to CPU type or operating system makes it an ideal display solution for a wide variety of applications.

FEATURES

- Embedded Display Buffer.
- Low Operating Voltage.
- Low-latency CPU interface.
- Direct support for the multiple CPU types.
- Programmable Resolutions and Color depths.
- STN LCD support.
- Active Matrix LCD support.
- Reflective Active Matrix support.
- SwivelView™ (hardware rotation of displayed image).
  - (Patent # 5,734,875 - Patent # 5,956,049)
- "Picture-in-Picture Plus".
- Ink Layer.
- Software Initiated Power Save Mode.
- Hardware or Software Video Invert.
- 120-pin PFBGA package.

SYSTEM BLOCK DIAGRAM



S1D13708

■ DESCRIPTION

Memory Interface

- Embedded 80K byte SRAM display buffer.

CPU Interface

- 'Fixed' low-latency CPU access times.
- Direct support for:
  Hitachi SH-4 / SH-3.
  Motorola M68xxx (DragonBall, ColdFire,REDCAP2).
  MPU bus interface with programmable READY.
- InDirect Interface
  provides a minimum 15-pin interface (as compared to 42-pin max implementation).

Display Support

- 4/8-bit monochrome LCD interface.
- 4/8/16-bit color STN LCD interface.
- Single-panel, single-drive passive displays.
- 9/12/18-bit Active matrix TFT interface.
- 'Direct' support for multiple TFT interfaces (Epson, Sharp, Type 2,3,4 external timing control IC not required).
- Typical resolutions supported (Ink Layer disabled):
  - 320x240 @ 8bpp
  - 160x160 @ 16bpp
  - 160x240 @ 16bpp

Power Down Modes

- Software Initiated Power Save Mode.
- BCLK can be switched off while maintaining LCD refresh.

Display Modes

- 1/2/4/8/16 bit-per-pixel (bpp) support.
- Up to 64 gray shades using FRM and dithering on monochrome passive LCD panels.
- Up to 64K colors on passive STN panels.
- Up to 64K colors on active matrix panels.
- SwivelView: direct hardware rotation of display image by 90°, 180°, 270°.
- "Picture-in-Picture Plus": displays a variable size window overlaid over background image.
- Ink Layer.
- Partial Display Support (available on Type 3 TFT).
- Double Buffering/multi-pages: provides smooth animation and instantaneous screen update.

Clock Source

- Two clock inputs (single clock possible).
- Clock source can be internally divided down for a higher frequency clock input.
- 12MHz Crystal Input.

Operating Voltage

- $CORE_{VDD}$ 1.8 to 2.2 volts.
- IO_{VDD} 3.0 to 3.6volts.

Package

- 120-pin PFBGA.

CONTACT YOUR SALES REPRESENTATIVE FOR THESE COMPREHENSIVE DESIGN TOOLS

- S1D13708 Technical Manual
- S5U13708 Evaluation Boards
- CPU Independent Software Utilities
- Palm OS® Hardware Abstraction Layer
- Windows® CE Display Driver
- VXWorks® Tornado™ Display Driver

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de

Taiwan
Epson Taiwan Technology & Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

Hardware Functional Specification

Document Number: X39A-A-001-02

THIS PAGE LEFT BLANK

Table of Contents

List of Tables

THIS PAGE LEFT BLANK

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

1.1 Scope

This is the Hardware Functional Specification for the S1D13708 Embedded Memory LCD Controller. Included in this document are timing diagrams, AC and DC characteristics, register descriptions, and power management descriptions. This document is intended for two audiences: Video Subsystem Designers and Software Developers.

For additional documentation related to the S1D13708 see Section 20, “References” on page 217.

This document is updated as appropriate. Please check the Epson Research and Development Website at **www.erd.epson.com** for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

1.2 Overview Description

The S1D13708 is a color/monochrome LCD graphics controller with an embedded 80K byte SRAM display buffer. While supporting all other panel types, the S1D13708 also directly interfaces to a variety of TFT products, thus removing the requirement of an external Timing Control IC. This high level of integration provides a low cost, low power, single chip solution to meet the demands of embedded markets such as Mobile Communications devices, and Palm-size PCs where board size and battery life are major concerns.

The S1D13708 utilizes a guaranteed low-latency CPU architecture providing support for microprocessors without READY/WAIT# handshaking signals. The 32-bit internal data path provides high performance bandwidth into display memory allowing for fast screen updates.

Products requiring a rotated display image can take advantage of the SwivelView ™ feature which provides hardware rotation of the display memory transparent to the software application. The S1D13708 also provides support for Virtual screen sizes and Picture-in-Picture Plus (variable size Overlay window).

The S1D13708’s impartiality to CPU type or operating system makes it an ideal display solution for a wide variety of applications.

2 Features

2.1 Integrated Frame Buffer

- Embedded 80K byte SRAM display buffer.

2.2 CPU Interface

- Direct support of the following interfaces:
  Generic MPU bus interface using WAIT# signal.
  Hitachi SH-3.
  Hitachi SH-4.
  Motorola M68K.
  Motorola MC68EZ328/MC68VZ328 DragonBall.
  Motorola “REDCAP2” - no WAIT# signal.
  Indirect Interface (Mode 68/Mode 80).
- 8-bit processor support with “glue logic”.
- “Fixed” low-latency CPU access times.
- Registers are memory-mapped - M/R# input selects between memory and register address space.
- The complete 80K byte display buffer is directly and contiguously available through the 17-bit address bus.
- Single level CPU write buffer.

2.3 Display Support

- 4/8-bit monochrome LCD interface.
- 4/8/16-bit color LCD interface.
- Single-panel, single-drive passive displays.
- 9/12/18-bit Active Matrix TFT interface.
- Direct support for 18-bit Epson D-TFD interface.
- Direct support for 18-bit Sharp HR-TFT interface.
- Direct support for 18-bit Type 2, 3, and 4 TFT interfaces.

2.4 Display Modes

- 1/2/4/8/16 bit-per-pixel (bpp) color depths.
- Up to 64 gray shades on monochrome passive LCD panels or 262144 colors on color passive LCD panels using Frame Rate Modulation (FRM) and dithering.
- Up to 64 gray shades or 262144 colors on active matrix LCD panels.
- Up to 64 gray shades or 256 colors can be simultaneously displayed in 8 bpp mode.
- Up to 64 gray shades or 65536 colors can be simultaneously displayed in 16 bpp mode.
- Example resolutions:
  320x240 at a color depth of 8 bpp
  160x160 at a color depth of 16 bpp
  160x240 at a color depth of 16 bpp
- Example resolutions with Ink Layer enabled.
  640x240 at a color depth of 2 bpp
  320x240 at a color depth of 4 bpp
  160x120 at a color depth of 16 bpp

2.5 Display Features

- SwivelView™: 90°, 180°, 270° counter-clockwise hardware rotation of display image.
- Virtual display support: displays images larger than the panel size through the use of panning and scrolling.
- Picture-in-Picture Plus: displays a variable size window overlaid over background image.
- Ink Layer.
- Double Buffering/Multi-pages: provides smooth animation and instantaneous screen updates.

2.6 Clock Source

- Three clock inputs: CLKI, CLKI2 and XTAL. It is possible to use one clock input only.
- Bus clock can be internally divided by 2, 3, or 4.
- Memory clock is derived from bus clock, CLKI2 or XTAL (XTAL is only available when configured for Indirect Interface). It can be internally divided by 2, 3, or 4.
- Pixel clock can be derived from CLKI, CLKI2, XTAL, bus clock, or memory clock. It can be internally divided by 2, 3, 4, or 8.

2.7 Operating Voltage

- CORE V_{DD} 1.62 to 1.98 volts.
- IO V_{DD} 3.0 to 3.6 volts.

2.8 Miscellaneous

- Hardware/Software Video Invert.
- Software Power Save mode.
- General Purpose Input/Output pins are available.
- BCLK can be switched off while still maintaining LCD refresh, offering power savings.
- 120-pin PFBGA package (also available in die form).
- 12MHz maximum crystal oscillator (XTAL) available for Indirect Interface.

3 Typical System Implementation Diagrams

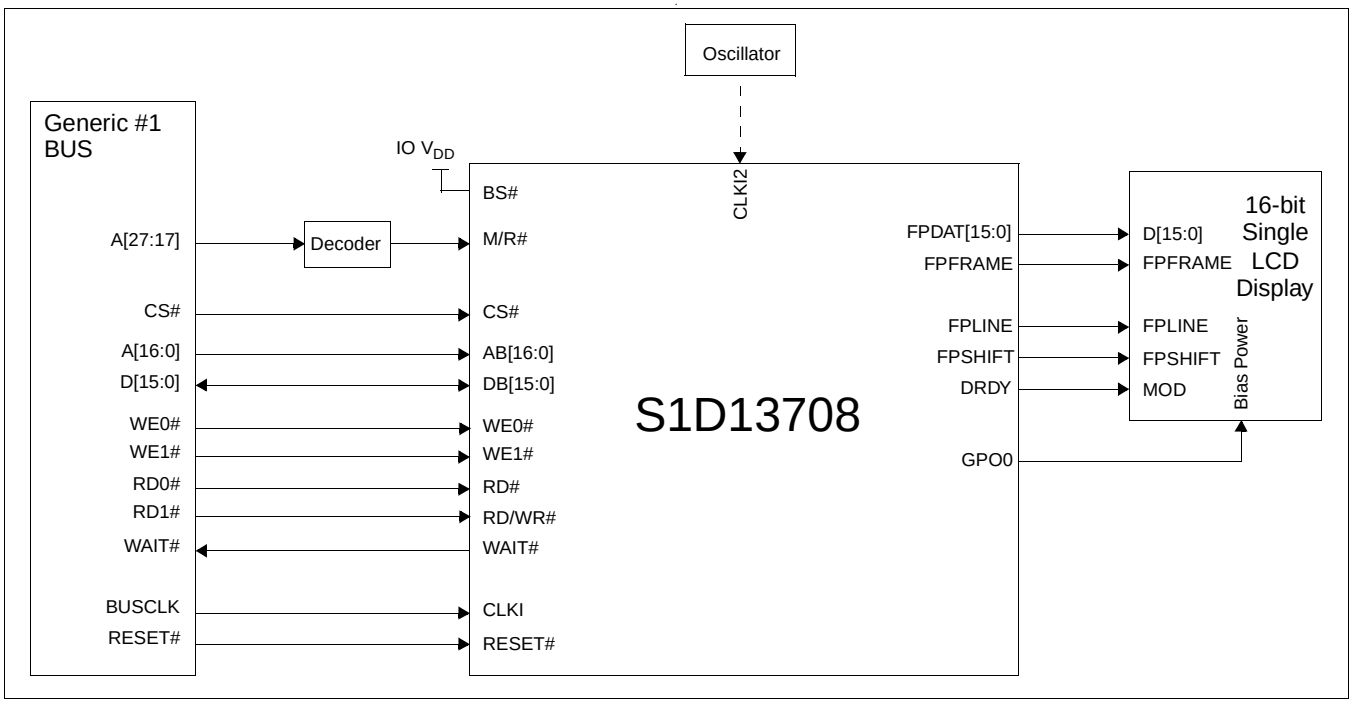


Figure 3-1 Typical System Diagram (Generic #1 Bus)

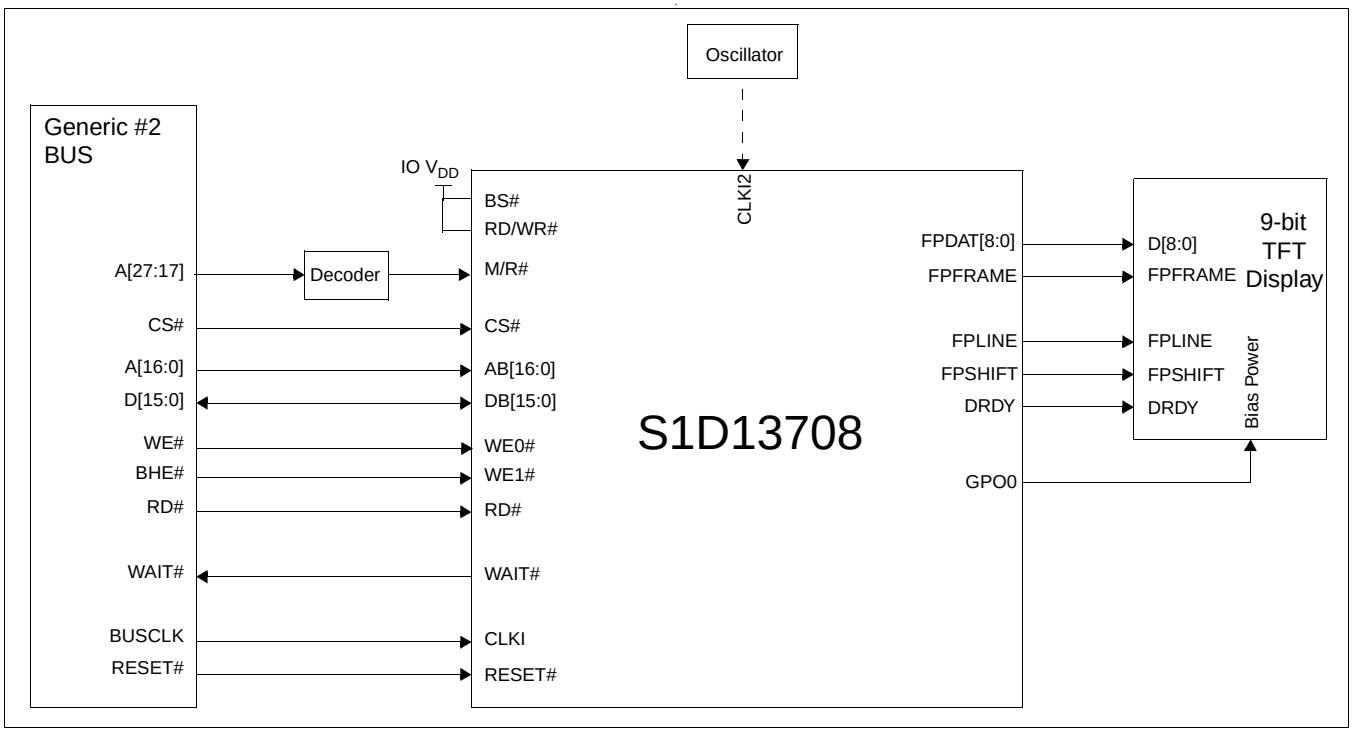


Figure 3-2 Typical System Diagram (Generic #2 Bus)

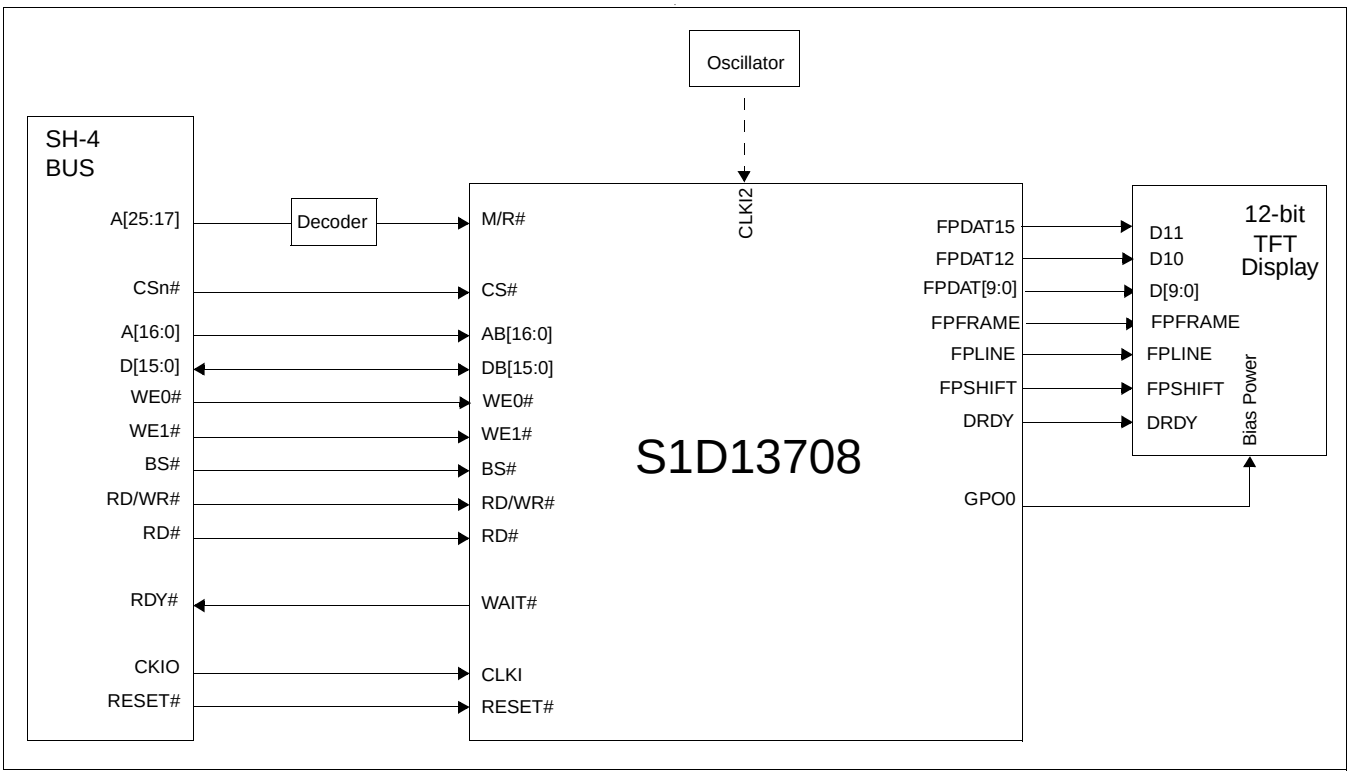


Figure 3-3 Typical System Diagram (Hitachi SH-4 Bus)

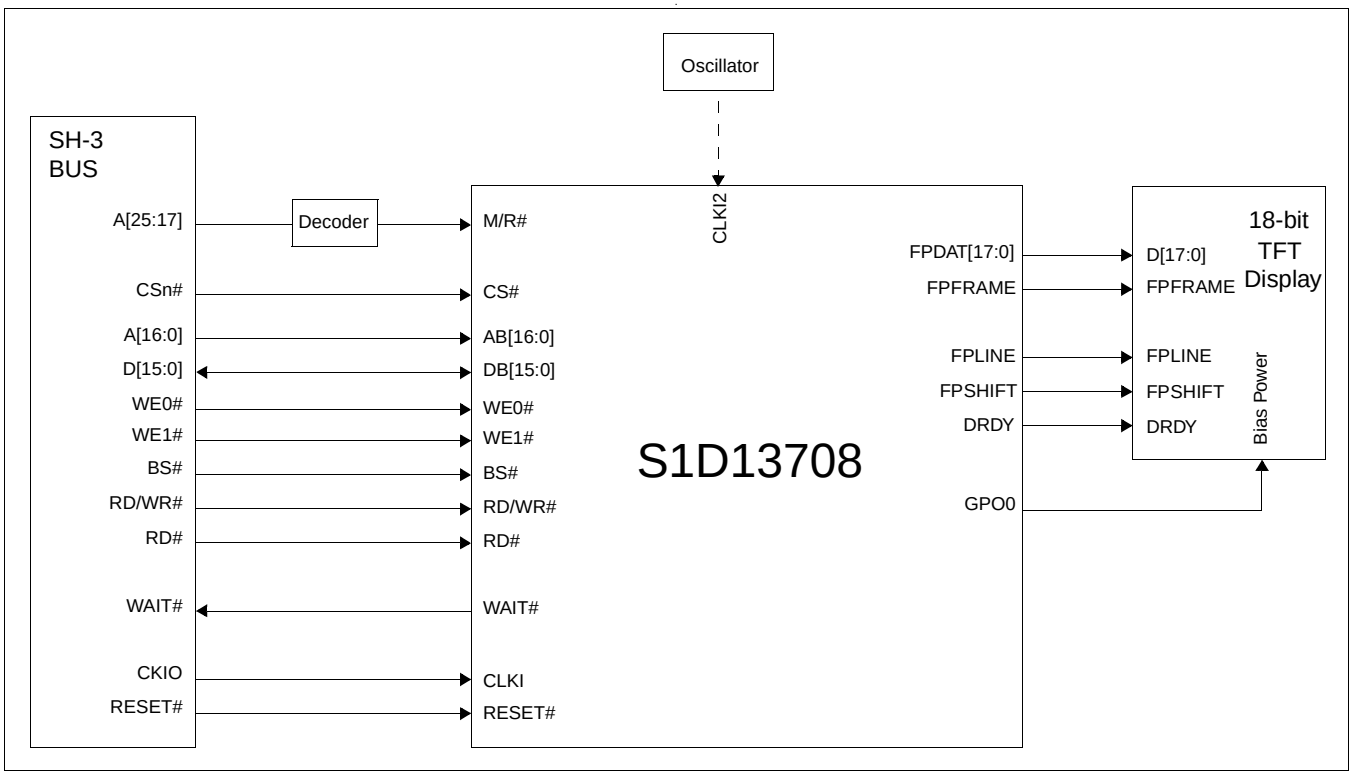


Figure 3-4 Typical System Diagram (Hitachi SH-3 Bus)

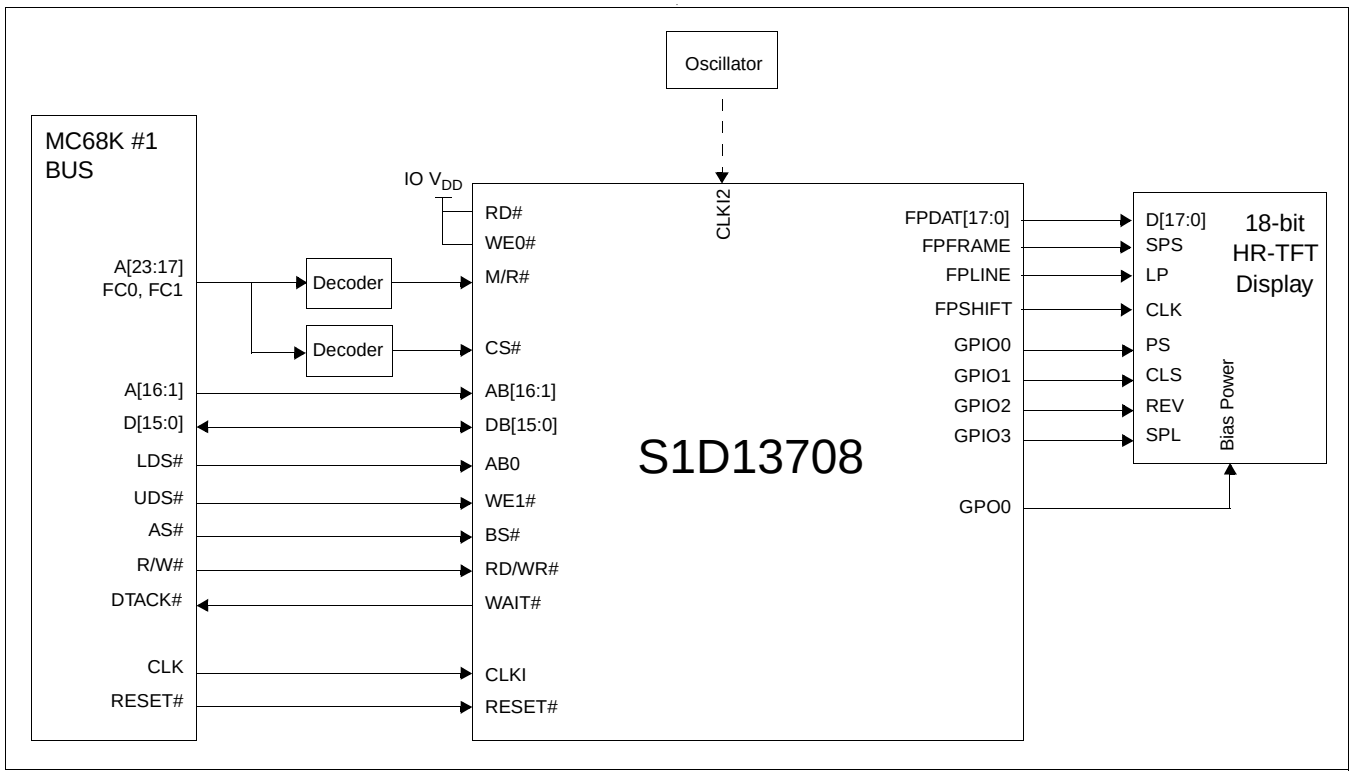


Figure 3-5 Typical System Diagram (MC68K # 1, Motorola 16-Bit 68000)

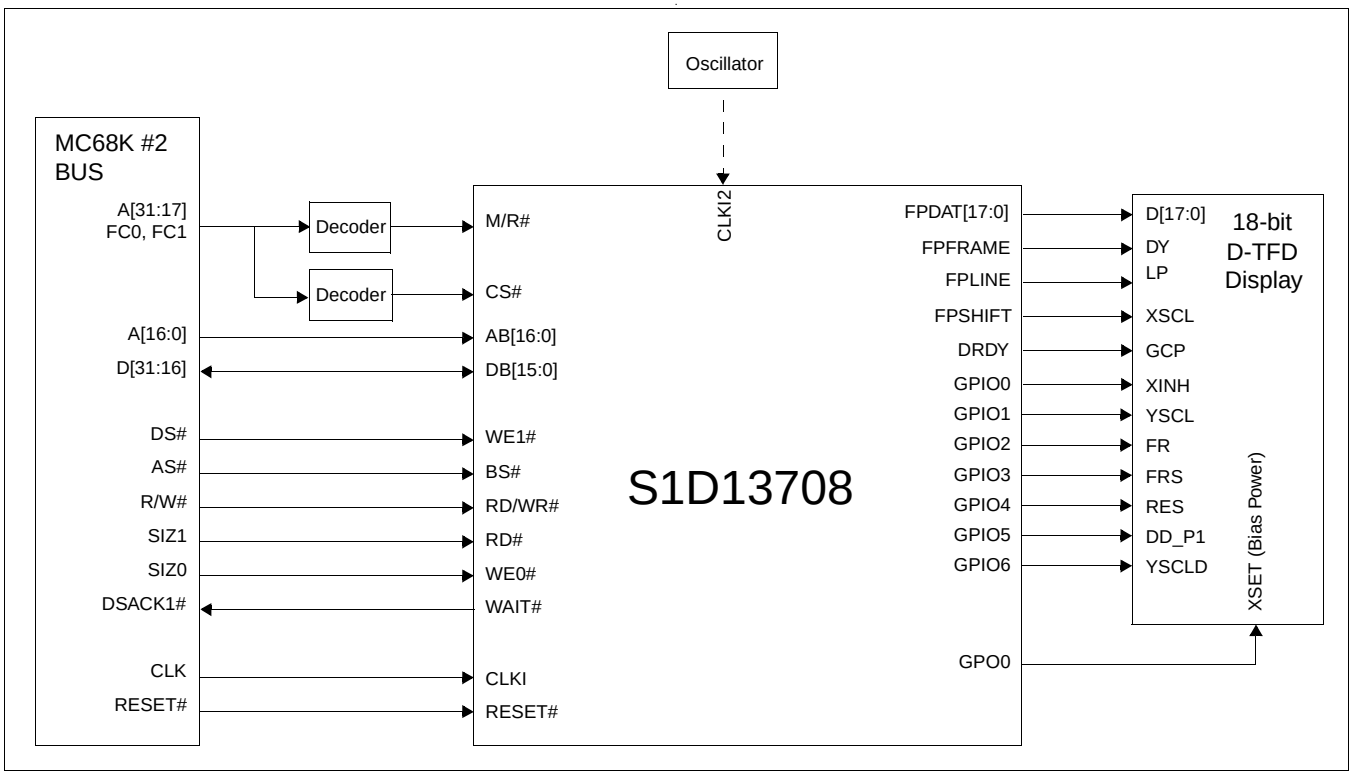


Figure 3-6 Typical System Diagram (MC68K #2, Motorola 32-Bit 68030)

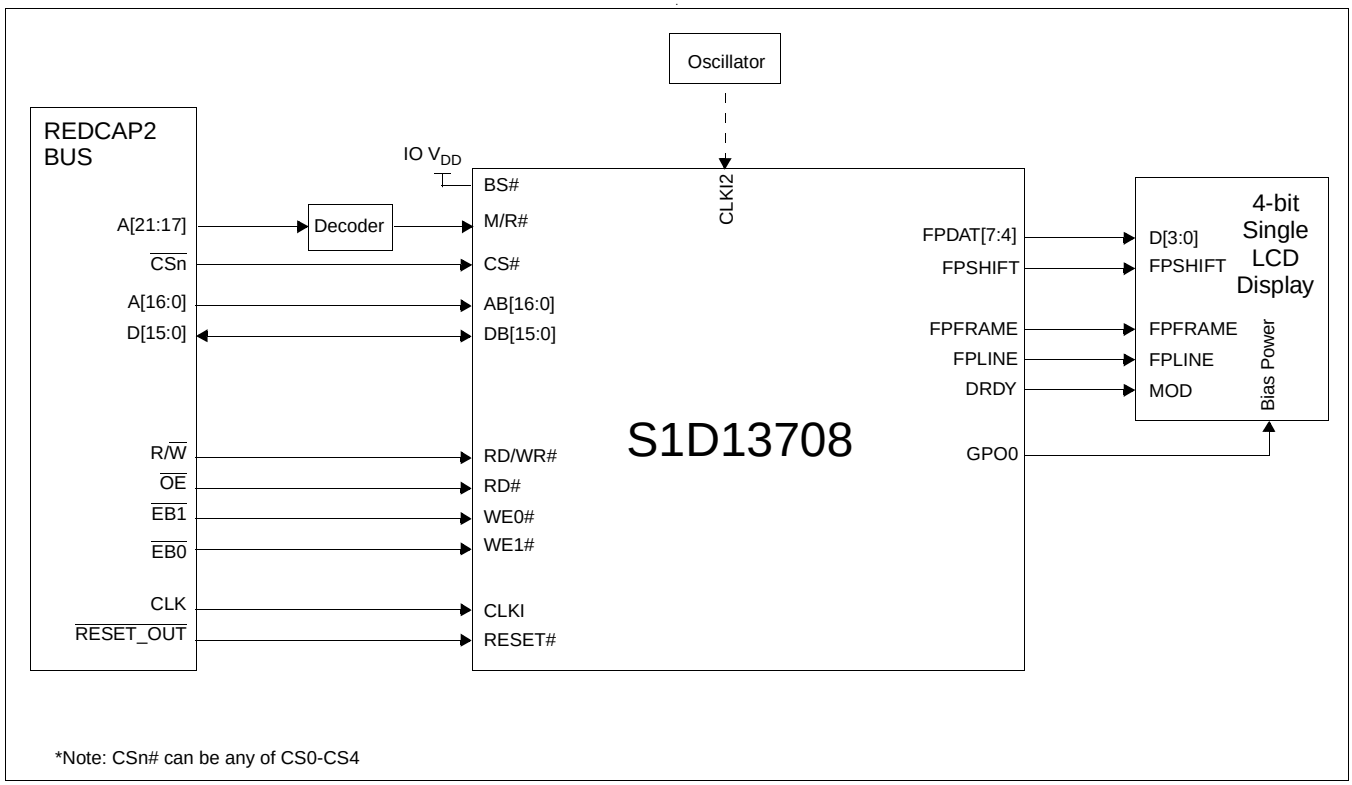


Figure 3-7 Typical System Diagram (Motorola REDCAP2 Bus)

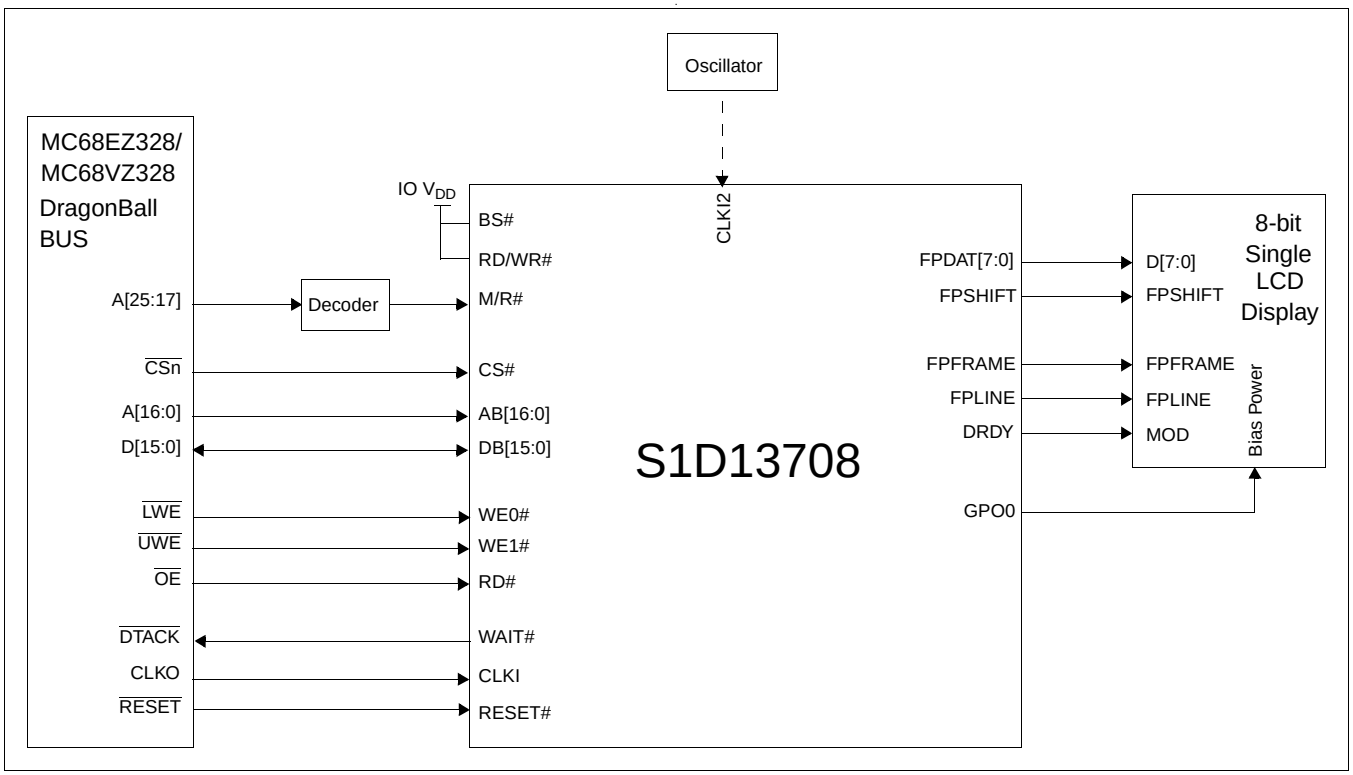


Figure 3-8 Typical System Diagram (Motorola MC68EZ328/MC68VZ328 "DragonBall" Bus)

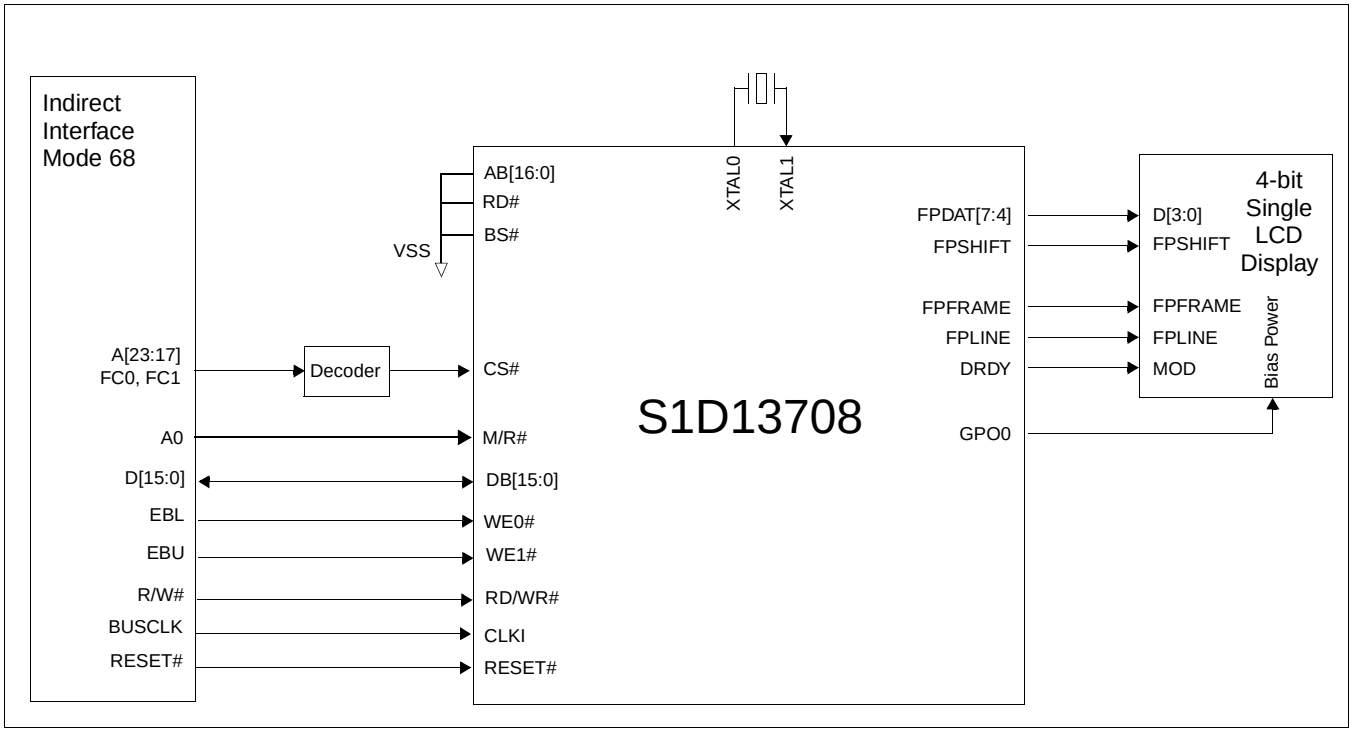


Figure 3-9 Typical System Diagram (Indirect Interface, Mode 68)

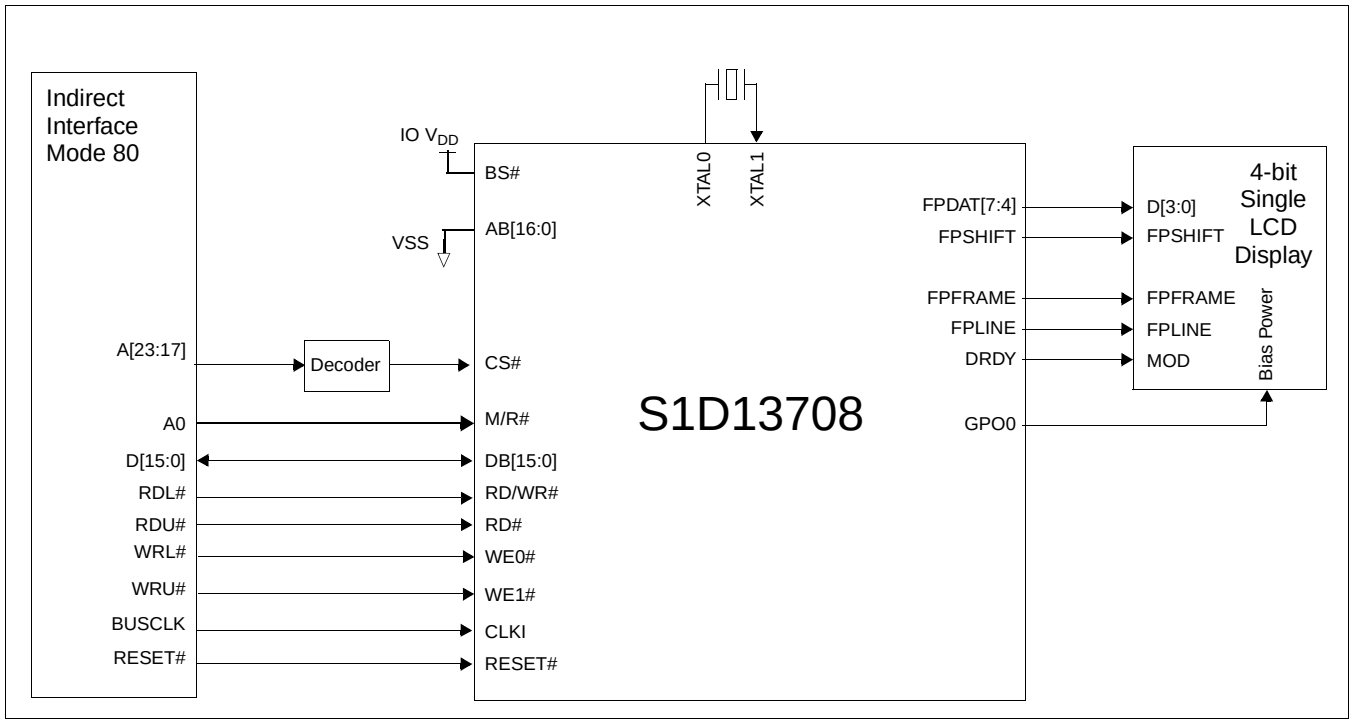


Figure 3-10 Typical System Diagram (Indirect Interface, Mode 80)

4 Pins

4.1 Pinout Diagram - PFBGA - 120pin

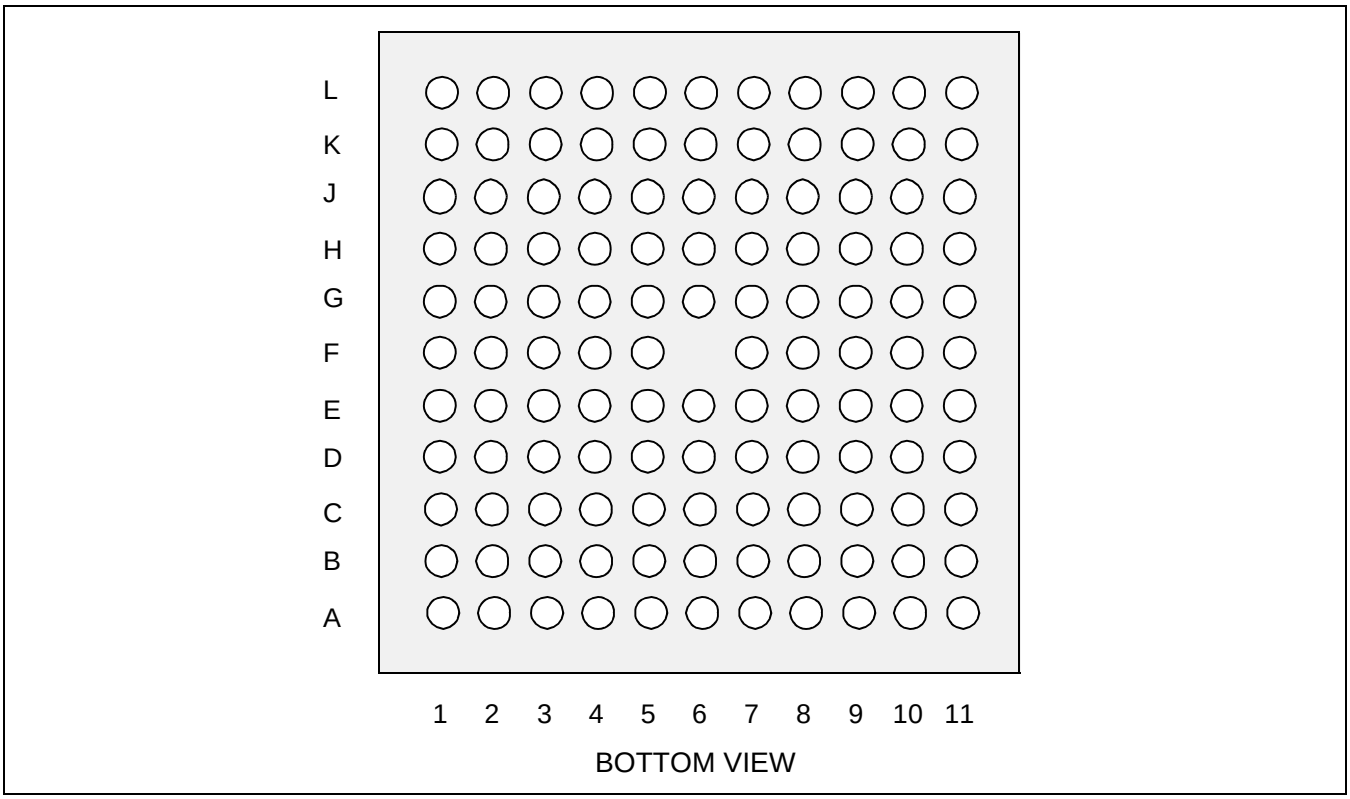


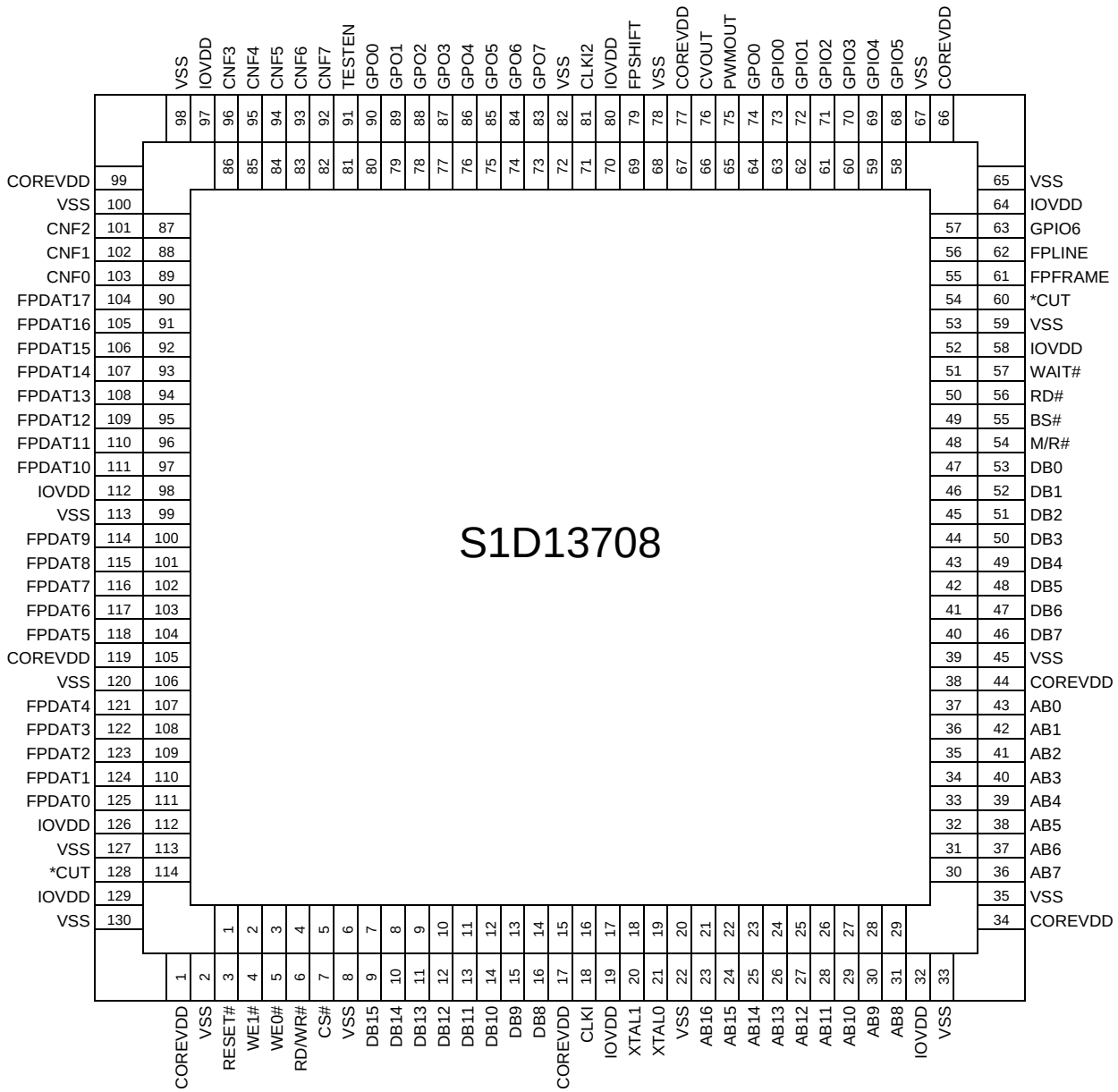
Figure 4-1 Pinout Diagram - PFBGA 120-pin

Table 4-1: PFBGA 120-pin Mapping

L	COREVDD	IOVDD	AB6	AB2	DB7	DB4	DB0	WAIT#	FPLINE	GPIO5	IOVDD
K	AB7	AB5	AB4	AB3	COREVDD	DB3	M/R#	IOVDD	GPIO6	GPIO4	COREVDD
J	AB10	AB9	AB8	AB1	DB6	DB2	BS#	FPFRAME	GPIO1	GPIO2	GPIO3
H	AB14	AB13	AB11	AB0	DB5	DB1	RD#	COREVDD	PWMOUT	GPIO0	DRDY
G	XTAL0	IOVDD	AB15	AB12	VSS	VSS	VSS	GPO6	CLKI2	FPSHIFT	CVOUT
F	COREVDD	CLKI	XTAL1	AB16	VSS		VSS	GPO2	GPO5	GPO7	IOVDD
E	DB11	DB10	DB8	VSS	VSS	VSS	VSS	CNF7	GPO1	GPO3	GPO4
D	DB15	DB14	DB12	DB9	FPDAT0	FPDAT6	FPDAT12	FPDAT16	CNF6	TESTEN	GPO0
C	WE0#	CS#	DB13	FPDAT1	FPDAT4	FPDAT7	IOVDD	FPDAT13	FPDAT17	CNF4	CNF5
B	COREVDD	WE1#	RD/WR#	FPDAT2	COREVDD	FPDAT8	FPDAT10	FPDAT14	CNF0	CNF2	CNF3
A	IOVDD	RESET#	IOVDD	FPDAT3	FPDAT5	FPDAT9	FPDAT11	FPDAT15	CNF1	COREVDD	IOVDD
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**

4.2 Pinout Diagram - Die Form

Table 4-2: S1D13708 Pad Layout



4.3 Pin Descriptions

Key:

I	=	Input
O	=	Output
IO	=	Bi-Directional (Input/Output)
P	=	Power pin
PCLKI1	=	CMOS/LVTTL schmitt input clock buffer
PIC	=	CMOS/LVTTL input buffer
PICS	=	CMOS/LVTTL input buffer with Schmitt input
POC8	=	CMOS/LVTTL 8mA low noise output buffer
PBCC8	=	CMOS/LVTTL bi-directional low noise buffer with 8mA CMOS output
PBCC8C	=	CMOS/LVTTL high-speed bi-directional low noise buffer with 8mA CMOS output
Hi-Z	=	High Impedance
POSC1	=	Crystal oscillator IO cell

4.3.1 Host Interface

Table 4-3: Host Interface Pin Descriptions

Pin Name	Type	PFBGA Pin #	Cell	IO Voltage	RESET # State	Description
AB0	I	H4	PIC	IOVDD	0	This input pin has multiple functions. • For Generic #1, this pin inputs system address bit 0 (A0). • For Generic #2, this pin inputs system address bit 0 (A0). • For SH-3/SH-4, this pin inputs system address bit 0 (A0). • For MC68K #1, this pin inputs the lower data strobe (LDS#). • For MC68K #2, this pin inputs system address bit 0 (A0). • For REDCAP2, this pin inputs system address bit 0 (A0). • For DragonBall, this pin inputs system address bit 0 (A0). • For Indirect (Mode 68), this pin is tied to V_{SS}. • For Indirect (Mode 80), this pin is tied to V_{SS}. See Table 4-9: “Host Bus Interface Pin Mapping,” on page 39 for summary.
AB[16:1]	I	F4,G3, G4,H1, H2,H3, J1,J2, J3,J4, K1,K2, K3,K4, L3,L4	PIC	IOVDD	0	System address bus bits 16-1. • For Generic #1, these pins input system address bits 16-1. • For Generic #2, these pins input system address bits 16-1. • For SH-3/SH-4, these pins input system address bits 16-1. • For MC68K #1, these pins input system address bits 16-1. • For MC68K #2, these pins input system address bits 16-1. • For REDCAP2, these pins input system address bits 16-1. • For DragonBall, these pins input system address bits 16-1. • For Indirect (Mode 68), these pins are tied to V_{SS}. • For Indirect (Mode 80), these pins are tied to V_{SS}.

Table 4-3: Host Interface Pin Descriptions

Pin Name	Type	PFBGA Pin #	Cell	IO Voltage	RESET # State	Description
DB[15:0]	IO	C3,D1, D2,D3, D4,E1, E2,E3, H5,H6, J5,J6, K6,L5, L6,L7	PBCC8	IOVDD	Hi-Z	Input data from the system data bus. • For Generic #1, these pins are connected to D[15:0]. • For Generic #2, these pins are connected to D[15:0]. • For SH-3/SH-4, these pins are connected to D[15:0]. • For MC68K #1, these pins are connected to D[15:0]. • For MC68K #2, these pins are connected to D[31:16] for a 32-bit device (e.g. MC68030) or D[15:0] for a 16-bit device (e.g. MC68340). • For REDCAP2, these pins are connected to D[15:0]. • For DragonBall, these pins are connected to D[15:0]. • For Indirect (Mode 68), these pins are connected to D[15:0]. • For Indirect (Mode 80), these pins are connected to D[15:0]. Unused pins should be tied to V_{SS}. See Table 4-9: “Host Bus Interface Pin Mapping,” on page 39 for summary.
WE0#	I	C1	PIC	IOVDD	1	This input pin has multiple functions. • For Generic #1, this pin inputs the write enable signal for the lower data byte (WE0#). • For Generic #2, this pin inputs the write enable signal (WE#) • For SH-3/SH-4, this pin inputs the write enable signal for data byte 0 (WE0#). • For MC68K #1, this pin must be tied to IO V_{DD} • For MC68K #2, this pin inputs the bus size bit 0 (SIZ0). • For REDCAP2, this pin inputs the byte enable signal for the D[7:0] data byte ($\overline{EB1}$). • For DragonBall, this pin inputs the byte enable signal for the D[7:0] data byte ($\overline{LWE}$). • For Indirect (Mode 68), this pin inputs the byte enable signal for the D[7:0] data byte (EBL). • For Indirect (Mode 80), this pin inputs the write enable signal for data byte 0 (WRL#). See Table 4-9: “Host Bus Interface Pin Mapping,” on page 39 for summary.

Table 4-3: Host Interface Pin Descriptions

Pin Name	Type	PFBGA Pin #	Cell	IO Voltage	RESET # State	Description
WE1#	I	B2	PIC	IOVDD	1	This input pin has multiple functions. • For Generic #1, this pin inputs the write enable signal for the upper data byte (WE1#). • For Generic #2, this pin inputs the byte enable signal for the high data byte (BHE#). • For SH-3/SH-4, this pin inputs the write enable signal for data byte 1 (WE1#). • For MC68K #1, this pin inputs the upper data strobe (UDS#). • For MC68K #2, this pin inputs the data strobe (DS#). • For REDCAP2, this pin inputs the byte enable signal for the D[15:8] data byte ($\overline{EB0}$). • For DragonBall, this pin inputs the byte enable signal for the D[15:8] data byte ($\overline{UWE}$). • For Indirect (Mode 68), this pin inputs the byte enable signal for the D[15:8] data byte (EBU). • For Indirect (Mode 80), this pin inputs the write enable signal for data byte 1 (WRU#). See Table 4-9: “Host Bus Interface Pin Mapping,” on page 39 for summary.
CS#	I	C2	PIC	IOVDD	1	Chip select input. See Table 4-9: “Host Bus Interface Pin Mapping,” on page 39 for summary.
M/R#	I	K7	PIC	IOVDD	0	This input pin is used to select between the display buffer and register address spaces of the S1D13708. M/R# is set high to access the display buffer and low to access the registers. See Table 4-9: “Host Bus Interface Pin Mapping,” on page 39 for summary.
BS#	I	J7	PIC	IOVDD	1	This input pin has multiple functions. • For Generic #1, this pin must be tied to IO V_{DD}. • For Generic #2, this pin must be tied to IO V_{DD}. • For SH-3/SH-4, this pin inputs the bus start signal (BS#). • For MC68K #1, this pin inputs the address strobe (AS#). • For MC68K #2, this pin inputs the address strobe (AS#). • For REDCAP2, this pin must be tied to IO V_{DD}. • For DragonBall, this pin must be tied to IO V_{DD}. • For Indirect (Mode 68), this pin selects the Indirect Interface For Mode 68, this pin is tied to V_{SS}. • For Indirect (Mode 80), this pin selects the Indirect Interface For Mode 80, this pin is tied to IO V_{DD}. See Table 4-9: “Host Bus Interface Pin Mapping,” on page 39 for summary.

Table 4-3: Host Interface Pin Descriptions

Pin Name	Type	PFBGA Pin #	Cell	IO Voltage	RESET # State	Description
RD/WR#	I	B3	PIC	IOVDD	1	This input pin has multiple functions. • For Generic #1, this pin inputs the read command for the upper data byte (RD1#). • For Generic #2, this pin must be tied to IO V_{DD}. • For SH-3/SH-4, this pin inputs the RD/WR# signal. The S1D13708 needs this signal for early decode of the bus cycle. • For MC68K #1, this pin inputs the R/W# signal. • For MC68K #2, this pin inputs the R/W# signal. • For REDCAP2, this pin inputs the R/$\overline{W}$ signal. • For DragonBall, this pin must be tied to IO V_{DD}. • For Indirect (Mode 68), this pin inputs the R/W# signal. • For Indirect (Mode 80, this pin inputs the read enable signal for data byte 0 (RDL#). See Table 4-9: “Host Bus Interface Pin Mapping,” on page 39 for summary.
RD#	I	H7	PIC	IOVDD	1	This input pin has multiple functions. • For Generic #1, this pin inputs the read command for the lower data byte (RD0#). • For Generic #2, this pin inputs the read command (RD#). • For SH-3/SH-4, this pin inputs the read signal (RD#). • For MC68K #1, this pin must be tied to IO V_{DD}. • For MC68K #2, this pin inputs the bus size bit 1 (SIZ1). • For REDCAP2, this pin inputs the output enable ($\overline{OE}$). • For DragonBall, this pin inputs the output enable ($\overline{OE}$). • For Indirect (Mode 68), this pin is tied to V_{SS}. • For Indirect (Mode 80), this pin inputs the read enable signal for data byte 1 (RDU#). See Table 4-9: “Host Bus Interface Pin Mapping,” on page 39 for summary.

Table 4-3: Host Interface Pin Descriptions

Pin Name	Type	PFBGA Pin #	Cell	IO Voltage	RESET # State	Description
WAIT#	O	L8	PBCC8C	IOVDD	Hi-Z	During a data transfer, this output pin is driven active to force the system to insert wait states. It is driven inactive to indicate the completion of a data transfer. WAIT# is released to the high impedance state after the data transfer is complete. Its active polarity is configurable. See Table 4-8: "Summary of Power-On/Reset Options," on page 38. • For Generic #1, this pin outputs the wait signal (WAIT#). • For Generic #2, this pin outputs the wait signal (WAIT#). • For SH-3 mode, this pin outputs the wait request signal (WAIT#). • For SH-4 mode, this pin outputs the device ready signal (RDY#). • For MC68K #1, this pin outputs the data transfer acknowledge signal (DTACK#). • For MC68K #2, this pin outputs the data transfer and size acknowledge bit 1 (DSACK1#). • For REDCAP2, this pin is unused (Hi-Z). • For DragonBall, this pin outputs the data transfer acknowledge signal (DTACK). • For Indirect (Mode 68), this pin is unused (Hi-Z). • For Indirect (Mode 80), this pin is unused (Hi-Z). See Table 4-9: "Host Bus Interface Pin Mapping," on page 39 for summary.
RESET#	I	A2	PICS	IOVDD	0	Active low input to set all internal registers to the default state and to force all signals to their inactive states.

4.3.2 LCD Interface

Table 4-4: LCD Interface Pin Descriptions

Pin Name	Type	PFBGA Pin #	Cell	IO Voltage	RESET# State	Description
FPDAT[17:0]	O	A4,A5, A6,A7, A8,B4, B6,B7, B8,C4, C5,C6, C8,C9, D5,D6, D7,D8	PBCC8	IOVDD	0	Panel Data bits 17-0.
FPFRAME	O	J8	PBCC8	IOVDD	0	This output pin has multiple functions. • Frame Pulse • SPS for Sharp HR-TFT • DY for Epson D-TFD • STV for TFT Type 2 • STV for Type 3 See Table 4-10: “LCD Interface Pin Mapping,” on page 40 for summary.
FPLINE	O	L9	PBCC8	IOVDD	0	This output pin has multiple functions. • Line Pulse • LP for Sharp HR-TFT • LP for Epson D-TFD • STB for TFT Type 2 • LP for Type 3 See Table 4-10: “LCD Interface Pin Mapping,” on page 40 for summary.
FPSHIFT	O	G10	PBCC8	IOVDD	0	This output pin has multiple functions. • Shift Clock • CLK for Sharp HR-TFT • XSCL for Epson D-TFD • CLK for TFT Type 2 • CPH for Type 3 See Table 4-10: “LCD Interface Pin Mapping,” on page 40 for summary.
DRDY	O	H11	POC8	IOVDD	0	This output pin has multiple functions. • Display enable (DRDY) for TFT panels • 2nd shift clock (FPSHIFT2) for passive LCD with Format 1 interface • GCP for Epson D-TFD • INV for TFT Type 2 • INV for TFT Type 3 • LCD backplane bias signal (MOD) for all other LCD panels See Table 4-10: “LCD Interface Pin Mapping,” on page 40 for summary.

Table 4-4: LCD Interface Pin Descriptions

Pin Name	Type	PFBGA Pin #	Cell	IO Voltage	RESET# State	Description
GPIO0	IO	H10	PBCC8	IOVDD	0	This pin has multiple functions. • PS for Sharp HR-TFT • XINH for Epson D-TFD • VCLK for TFT Type 2 • CPV for Type 3 • General purpose IO pin 0 (GPIO0) • Hardware Video Invert See Table 4-10: “LCD Interface Pin Mapping,” on page 40 for summary.
GPIO1	IO	J9	PBCC8	IOVDD	0	This pin has multiple functions. • CLS for Sharp HR-TFT • YSCL for Epson D-TFD • AP for TFT Type 2 • OE for Type 3 • General purpose IO pin 1 (GPIO1) See Table 4-10: “LCD Interface Pin Mapping,” on page 40 for summary.
GPIO2	IO	J10	PBCC8	IOVDD	0	This pin has multiple functions. • REV for Sharp HR-TFT • FR for Epson D-TFD • POL for TFT Type 2 • POL for Type 3 • General purpose IO pin 2 (GPIO2) See Table 4-10: “LCD Interface Pin Mapping,” on page 40 for summary.
GPIO3	IO	J11	PBCC8	IOVDD	0	This pin has multiple functions. • SPL for Sharp HR-TFT • FRS for Epson D-TFD • STH for TFT Type 2 • EIO for Type 3 • General purpose IO pin 3 (GPIO3) See Table 4-10: “LCD Interface Pin Mapping,” on page 40 for summary.
GPIO4	IO	K10	PBCC8	IOVDD	0	This pin has multiple functions. • RES for Epson D-TFD • VCOM for Type 3 • General purpose IO pin 4 (GPIO4) See Table 4-10: “LCD Interface Pin Mapping,” on page 40 for summary.

Table 4-4: LCD Interface Pin Descriptions

Pin Name	Type	PFBGA Pin #	Cell	IO Voltage	RESET# State	Description
GPIO5	IO	L10	PBCC8	IOVDD	0	This pin has multiple functions. • DD_P1 for Epson D-TFD • XOEV for Type 3 • General purpose IO pin 5 (GPIO5) See Table 4-10: “LCD Interface Pin Mapping,” on page 40 for summary.
GPIO6	IO	K9	PBCC8	IOVDD	0	This pin has multiple functions. • YSCLD for Epson D-TFD • CMD for Type 3 • General purpose IO pin 6 (GPIO6) See Table 4-10: “LCD Interface Pin Mapping,” on page 40 for summary.
PWMOUT	O	H9	PBCC8	IOVDD	0	This output pin has multiple functions. • PWM Clock output • General purpose output
CVOUT	O	G11	PBCC8	IOVDD	0	This output pin has multiple functions. • CV Pulse Output • General purpose output
GPO0	O	D11	POC8	IOVDD	0	General Purpose Output (possibly used for controlling the LCD power). It may also be used for the MOD control signal of the Sharp HR-TFT panel.
GPO1	O	E9	POC8	IOVDD	0	General Purpose Output pin 1 (GPO1). When used with a Type 3 panel this pin functions as PCLK1.
GPO2	O	F8	POC8	IOVDD	0	General Purpose Output pin 2 (GPO2). When used with a Type 3 panel this pin functions as PCLK2.
GPO3	O	E10	POC8	IOVDD	0	General Purpose Output pin 3 (GPO3). When used with a Type 3 panel this pin functions as XRESH.
GPO4	O	E11	POC8	IOVDD	0	General Purpose Output pin 4 (GPO4). When used with a Type 3 panel this pin functions as XRESV.
GPO5	O	F9	POC8	IOVDD	0	General Purpose Output pin 5 (GPO5). When used with a Type 3 panel this pin functions as XOHV.
GPO6	O	G8	POC8	IOVDD	0	General Purpose Output pin 6 (GPO6). When used with a Type 3 panel this pin functions as XSTBY.
GPO7	O	F10	POC8	IOVDD	0	General Purpose Output pin 7 (GPO7). When used with a Type 3 panel this pin functions as PMDE.

4.3.3 Clock Input

Table 4-5: Clock Input Pin Descriptions

Pin Name	Type	PFBGA Pin #	Cell	IO Voltage	RESET# State	Description
CLKI	I	F2	PCLK1	IOVDD	—	Typically used as input clock source for bus clock and memory clock.
CLKI2	I	G9	PCLK1	IOVDD	—	Typically used as input clock source for pixel clock This pin must be tied to V_{SS} when a crystal is not used.
XTAL0	0	F3	POSC1	IOVDD	—	Crystal output. This pin must be left unconnected when a crystal is not used.
XTAL1	I	G1	POSC1	IOVDD	—	Crystal input. This pin must be tied to V_{SS} when a crystal is not used.

4.3.4 Miscellaneous

Table 4-6: Miscellaneous Pin Descriptions

Pin Name	Type	PFBGA Pin #	Cell	IO Voltage	RESET# State	Description
CNF[7:0]	I	A9, B9, B10, B11, C10, C11, D9, E8	PIC	IOVDD	—	These inputs are used to configure the S1D13708 - see Table 4-8: “Summary of Power-On/Reset Options,” on page 38. **Note: These pins are used for configuration of the S1D13708 and must be connected directly to IO V_{DD} or V_{SS}.**
TESTEN	I	D10	PIC	IOVDD	0	Test Enable input used for production test only (has type 1 pull-down resistor with a typical value of 50Ω at 3.3V).

4.3.5 Power And Ground

Table 4-7: Power And Ground Pin Descriptions

Pin Name	Type	PFBGA Pin #	Cell	IO Voltage	RESET# State	Description
IOVDD	P	A1, A3, A11, C7, F11, G2, K8,L2, L11	P	—	—	IO V_{DD} pins.
COREVDD	P	A10, B1, B5, F1, H8, K5, K11, L1	P	—	—	Core $V_{DD.}$ pins.
VSS	P	E4, E5, E6, E7, F5, F7, G5, G6, G7	P	—	—	V_{SS} pins.

4.4 Summary of Configuration Options

Table 4-8: Summary of Power-On/Reset Options

<table>
<tr><th rowspan="2">S1D13708 Configuration Input</th><th colspan="2">Power-On/Reset State</th></tr>
<tr><th>1</th><th>0</th></tr>
<tr><td>CNF4,CNF[2:0]</td><td colspan="2">Select host bus interface as follows:
<table><tr><th>CNF4</th><th>CNF2</th><th>CNF1</th><th>CNF0</th><th>Host Bus</th></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>SH-4/SH-3 interface, Big Endian</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>SH-4/SH-3 interface, Little Endian</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>MC68K #1, Big Endian</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Reserved</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>MC68K #2, Big Endian</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Reserved</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>Generic #1, Big Endian</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>Generic #1, Little Endian</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>Reserved</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>Generic #2, Little Endian</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>REDCAP2, Big Endian</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>Reserved</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>DragonBall (MC68EZ328/MC68VZ328), Big Endian</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>Reserved</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>Indirect Interface, Big Endian</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>Indirect Interface, Little Endian</td></tr></table></td></tr>
<tr><td>CNF3</td><td>Configure GPIO pins as inputs at power-on</td><td>Configure GPIO pins as outputs at power-on (for use by HR-TFT/D-TFD/TFT Type 2/TFT Type 3 when selected)</td></tr>
<tr><td>CNF5</td><td>WAIT# is active high</td><td>WAIT# is active low</td></tr>
<tr><td>CNF[7:6]</td><td colspan="2">BCLK Source (CLKI/XTAL) to BCLK divide select:
Note: XTAL should only be used when configured for Indirect Interface (CNF[2:0] = 111).
<table><tr><th>CNF7</th><th>CNF6</th><th>BCLK Divide Ratio</th></tr><tr><td>0</td><td>0</td><td>1 : 1</td></tr><tr><td>0</td><td>1</td><td>2 : 1</td></tr><tr><td>1</td><td>0</td><td>3 : 1</td></tr><tr><td>1</td><td>1</td><td>4 : 1</td></tr></table></td></tr>
</table>

4.5 Host Bus Interface Pin Mapping

Table 4-9: Host Bus Interface Pin Mapping

S1D13708 Pin Name	Generic #1	Generic #2	Hitachi SH-3 /SH-4	Motorola MC68K #1	Motorola MC68K #2	Motorola REDCAP2	Motorola MC68EZ328/ MC68VZ328 DragonBall	Indirect Interface Mode 68	Indirect Interface Mode 80
AB[16:1]	A[16:1]	A[16:1]	A[16:1]	A[16:1]	A[16:1]	A[16:1]	A[16:1]	Connected to V_{SS}	
AB0	A0[1]	A0	A0[1]	LDS#	A0	A0[1]	A0[1]	Connected to V_{SS}	
DB[15:0]	D[15:0]	D[15:0]	D[15:0]	D[15:0]	D[15:0][2]	D[15:0]	D[15:0]	D[15:0]	D[15:0]
CS#	External Decode		CSn#	External Decode		$\overline{CSn}$	$\overline{CSn}$	External Decode	
M/R#	External Decode							A0	A0
CLKI	BUSCLK	BUSCLK	CKIO	CLK	CLK	CLK	CLKO	BUSCLK	BUSCLK
BS#	Connected to IO V_{DD}		BS#	AS#	AS#	Connected to IO V_{DD}		Connected to V_{SS}	Connected to IO V_{DD}
RD/WR#	RD1#	Connected to IO V_{DD}	RD/WR#	R/W#	R/W#	R/$\overline{W}$	Connected to IO V_{DD}	R/W#	RDL#
RD#	RD0#	RD#	RD#	Connected to IO V_{DD}	External Decode	$\overline{OE}$	$\overline{OE}$	Connected to V_{SS}	RDU#
WE0#	WE0#	WE#	WE0#	Connected to IO V_{DD}	SIZ0	$\overline{EB1}$	$\overline{LWE}$	EBL	WRL#
WE1#	WE1#	BHE#	WE1#	UDS#	DS#	$\overline{EB0}$	$\overline{UWE}$	EBU	WRU#
WAIT#	WAIT#	WAIT#	WAIT#/ RDY#	DTACK#	DSACK1#	N/A	$\overline{DTACK}$	N/A	N/A
RESET#	RESET#	RESET#	RESET#	RESET#	RESET#	$\overline{RESET\_OUT}$	$\overline{RESET}$	RESET#	RESET#

Note

[1] A0 for these busses is not used internally by the S1D13708.

[2] If the target MC68K bus is 32-bit, then these signals should be connected to D[31:16].

4.6 LCD Interface Pin Mapping

Table 4-10: LCD Interface Pin Mapping

Pin Name	Monochrome Passive Panel		Color Passive Panel				Color TFT Panel							
	Single		Single				Generic TFT (TFT Type 1)			Sharp HR-TFT[1]	Epson D-TFD[1]	TFT Type 2	TFT Type 3	TFT Type 4
	4-bit	8-bit	4-bit	Format 1 8-bit	Format 2 8-bit	16-Bit	9-bit	12-bit	18-bit	18-bit	18-bit	18-bit	18-bit	18-bit
FPFRAME	FPFRAME									SPS	DY	STV	STV	FPFRAME
FPLINE	FPLINE									LP	LP	STB	LP	FPLINE
FPSHIFT	FPSHIFT									DCLK	XSCL	CLK	CPH	FPSHIFT
DRDY	MOD			FPSHIFT2	MOD		DRDY			no connect	GCP	INV	INV	DRDY
FPDAT0	driven 0	D0	driven 0	D0 (B5)[2]	D0 (G3)[2]	D0 (R6)[2]	R2	R3	R5	R5	R5	R5	R5	R5
FPDAT1	driven 0	D1	driven 0	D1 (R5)[2]	D1 (R3)[2]	D1 (G5)[2]	R1	R2	R4	R4	R4	R4	R4	R4
FPDAT2	driven 0	D2	driven 0	D2 (G4)[2]	D2 (B2)[2]	D2 (B4)[2]	R0	R1	R3	R3	R3	R3	R3	R3
FPDAT3	driven 0	D3	driven 0	D3 (B3)[2]	D3 (G2)[2]	D3 (R4)[2]	G2	G3	G5	G5	G5	G5	G5	G5
FPDAT4	D0	D4	D0 (R2)[2]	D4 (R3)[2]	D4 (R2)[2]	D8 (B5)[2]	G1	G2	G4	G4	G4	G4	G4	G4
FPDAT5	D1	D5	D1 (B1)[2]	D5 (G2)[2]	D5 (B1)[2]	D9 (R5)[2]	G0	G1	G3	G3	G3	G3	G3	G3
FPDAT6	D2	D6	D2 (G1)[2]	D6 (B1)[2]	D6 (G1)[2]	D10 (G4)[2]	B2	B3	B5	B5	B5	B5	B5	B5
FPDAT7	D3	D7	D3 (R1)[2]	D7 (R1)[2]	D7 (R1)[2]	D11 (B3)[2]	B1	B2	B4	B4	B4	B4	B4	B4
FPDAT8	driven 0	driven 0	driven 0	driven 0	driven 0	D4 (G3)[2]	B0	B1	B3	B3	B3	B3	B3	B3
FPDAT9	driven 0	driven 0	driven 0	driven 0	driven 0	D5 (B2)[2]	driven 0	R0	R2	R2	R2	R2	R2	R2
FPDAT10	driven 0	driven 0	driven 0	driven 0	driven 0	D6 (R2)[2]	driven 0	driven 0	R1	R1	R1	R1	R1	R1
FPDAT11	driven 0	driven 0	driven 0	driven 0	driven 0	D7 (G1)[2]	driven 0	driven 0	R0	R0	R0	R0	R0	R0
FPDAT12	driven 0	driven 0	driven 0	driven 0	driven 0	D12 (R3)[2]	driven 0	G0	G2	G2	G2	G2	G2	G2
FPDAT13	driven 0	driven 0	driven 0	driven 0	driven 0	D13 (G2)[2]	driven 0	driven 0	G1	G1	G1	G1	G1	G1
FPDAT14	driven 0	driven 0	driven 0	driven 0	driven 0	D14 (B1)[2]	driven 0	driven 0	G0	G0	G0	G0	G0	G0
FPDAT15	driven 0	driven 0	driven 0	driven 0	driven 0	D15 (R1)[2]	driven 0	B0	B2	B2	B2	B2	B2	B2
FPDAT16	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	B1	B1	B1	B1	B1	B1
FPDAT17	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	driven 0	B0	B0	B0	B0	B0	B0
GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	GPIO0	PS	XINH	VCLK	CPV	GPIO0
GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	GPIO1	CLS	YSCL	AP	OE	GPIO1
GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	GPIO2	REV	FR	POL	POL	GPIO2
GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	GPIO3	SPL	FRS	STH	EIO	GPIO3
GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4	GPIO4 (output only)	RES	GPIO4 (output only)	VCOM	GPIO4
GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5	GPIO5 (output only)	DD_P1	GPIO5 (output only)	XOEV	GPIO5
GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6	GPIO6 (output only)	YSCLD	GPIO6 (output only)	CMD	GPIO6
GPO0	GPO0 (General Purpose Output)									MOD	GPO0	GPO0	GPO0	GPO0
GPO1	GPO1											GPO1	PCLK1	GPO1
GPO2	GPO2											GPO2	PCLK2	GPO2
GPO3	GPO3											GPO3	XRESH	GPO3
GPO4	GPO4											GPO4	XRESV	GPO4
GPO5	GPO5											GPO5	XOHV	GPO5
GPO6	GPO6											GPO6	XSTBY	GPO6
GPO7	GPO7											GPO7	PMDE	GPO7
CVOUT	CVOUT											CVOUT	CVOUT	CVOUT
PWMOUT	PWMOUT											PWMOUT	PWMOUT	PWMOUT

Note

[1] GPIO pins must be configured as outputs (CNF3 = 0 at RESET#) when TFT-Type 2, TFT-Type 3, HR-TFT or D-TFD panels are selected.

[2] These pin mappings use signal names commonly used for each panel type, however signal names may differ between panel manufacturers. The values shown in brackets represent the color components as mapped to the corresponding FPDATxx signals at the first valid edge of FPSHIFT. For further FPDATxx to LCD interface mapping, see Section 6.4, “Display Interface” on page 68.

5 D.C. Characteristics

Note

When applying Supply Voltages to the S1D13708, Core V_{DD} **must** be applied to the chip before, or simultaneously with IO V_{DD}, or damage to the chip may result.

Table 5-1: Absolute Maximum Ratings (Preliminary - Subject to Change)

Symbol	Parameter	Rating	Units
Core V_{DD}	Supply Voltage	V_{SS} - 0.3 to 4.0	V
IO V_{DD}	Supply Voltage	V_{SS} - 0.3 to 4.0	V
V_{IN}	Input Voltage	V_{SS} - 0.3 to IO V_{DD} + 0.5	V
V_{OUT}	Output Voltage	V_{SS} - 0.3 to IO V_{DD} + 0.5	V
T_{STG}	Storage Temperature	-65 to 150	° C
T_{SOL}	Solder Temperature/Time	260 for 10 sec. max at lead	° C

Table 5-2: Recommended Operating Conditions

Symbol	Parameter	Condition	Min	Typ	Max	Units
Core V_{DD}	Supply Voltage	V_{SS} = 0 V	1.62	1.8	1.98	V
IO V_{DD}	Supply Voltage	V_{SS} = 0 V	3.0	3.3	3.6	V
V_{IN}	Input Voltage		V_{SS}		IO V_{DD}	V
T_{OPR}	Operating Temperature		0		70	° C

Table 5-3: Electrical Characteristics for VDD = 3.3V typical

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{DDS}	Quiescent Current	Quiescent Conditions			170	µA
I_{IZ}	Input Leakage Current		-1		1	µA
I_{OZ}	Output Leakage Current		-1		1	µA
V_{OH}	High Level Output Voltage	VDD = min. I_{OH} = -8mA	V_{DD} - 0.4			V
V_{OL}	Low Level Output Voltage	VDD = min. I_{OL} = 8mA			0.4	V
V_{IH}	High Level Input Voltage	LVTTL Level, V_{DD} = max	2.0			V
V_{IL}	Low Level Input Voltage	LVTTL Level, V_{DD} = min			0.8	V
V_{T+}	High Level Input Voltage	LVTTL Schmitt	1.1		2.4	V
V_{T-}	Low Level Input Voltage	LVTTL Schmitt	0.6		1.8	V
V_{H1}	Hysteresis Voltage	LVTTL Schmitt	0.1			V
R_{PD}	Pull Down Resistance	$V_I = V_{DD}$	20	50	120	kΩ
C_I	Input Pin Capacitance				10	pF
C_O	Output Pin Capacitance				10	pF
C_{IO}	Bi-Directional Pin Capacitance				10	pF

6 A.C. Characteristics

Conditions: CORE V_{DD} = 1.8V ± 10%
IO V_{DD} = 3.3V ± 10%
T_A = TBD ° C
t_r and t_f for all inputs must be ≤ 5 nsec (10% ~ 90%)
C_L = 50pF (Bus/MPU Interface)
C_L = 0pF (LCD Panel Interface)

6.1 Clock Timing

6.1.1 Input Clocks

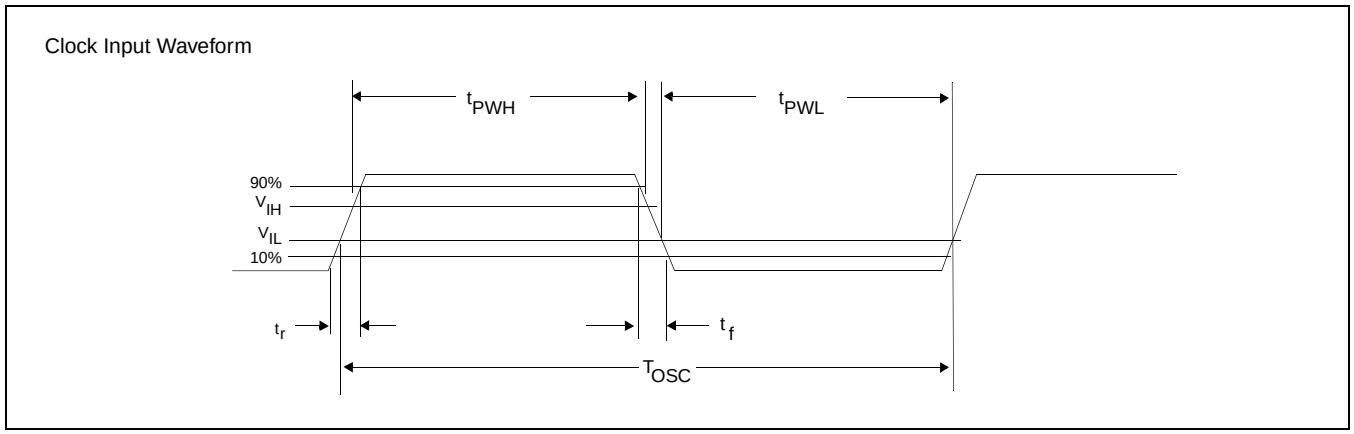


Figure 6-1 Clock Input Requirements

Table 6-1: Clock Input Requirements for CLKI when CLKI to BCLK divide > 1

Symbol	Parameter	Min	Max	Units
f_{OSC}	Input Clock Frequency (CLKI)		100	MHz
T_{OSC}	Input Clock period (CLKI)	$1/f_{OSC}$		ns
t_{PWH}	Input Clock Pulse Width High (CLKI)	4.5		ns
t_{PWL}	Input Clock Pulse Width Low (CLKI)	4.5		ns
t_f	Input Clock Fall Time (10% - 90%)		5	ns
t_r	Input Clock Rise Time (10% - 90%)		5	ns

Note

Maximum internal requirements for clocks derived from CLKI must be considered when determining the frequency of CLKI. See Section 6.1.2, “Internal Clocks” on page 43 for internal clock requirements.

Table 6-2: Clock Input Requirements for CLKI when CLKI to BCLK divide = 1

Symbol	Parameter	Min	Max	Units
f_{OSC}	Input Clock Frequency (CLKI)		66	MHz
T_{OSC}	Input Clock period (CLKI)	$1/f_{OSC}$		ns
t_{PWH}	Input Clock Pulse Width High (CLKI)	3		ns
t_{PWL}	Input Clock Pulse Width Low (CLKI)	3		ns
t_f	Input Clock Fall Time (10% - 90%)		5	ns
t_r	Input Clock Rise Time (10% - 90%)		5	ns

Note
Maximum internal requirements for clocks derived from CLKI must be considered when determining the frequency of CLKI. See Section 6.1.2, “Internal Clocks” on page 43 for internal clock requirements.

Table 6-3: Clock Input Requirements for CLKI2

Symbol	Parameter	Min	Max	Units
f_{OSC}	Input Clock Frequency (CLKI2)		66	MHz
T_{OSC}	Input Clock period (CLKI2)	$1/f_{OSC}$		ns
t_{PWH}	Input Clock Pulse Width High (CLKI2)	3		ns
t_{PWL}	Input Clock Pulse Width Low (CLKI2)	3		ns
t_f	Input Clock Fall Time (10% - 90%)		5	ns
t_r	Input Clock Rise Time (10% - 90%)		5	ns

Note
Maximum internal requirements for clocks derived from CLKI2 must be considered when determining the frequency of CLKI2. See Section 6.1.2, “Internal Clocks” on page 43 for internal clock requirements.

6.1.2 Internal Clocks

Table 6-4: Internal Clock Requirements

Symbol	Parameter	Min	Max	Units
f_{BCLK}	Bus Clock frequency		66	MHz
f_{MCLK}	Memory Clock frequency		50	MHz
f_{PCLK}	Pixel Clock frequency		50	MHz
f_{PWMCLK}	PWM Clock frequency		66	MHz
f_{XTAL}	XTAL Clock frequency		12	MHz

Note
For further information on internal clocks, refer to Section 7, “Clocks” on page 112.

6.2 CPU Interface Timing

The following section includes CPU interface AC Timing. These timings are based on IO V_{DD} = 3.3V and Core V_{DD} = 1.8V.

6.2.1 Generic #1 Interface Timing

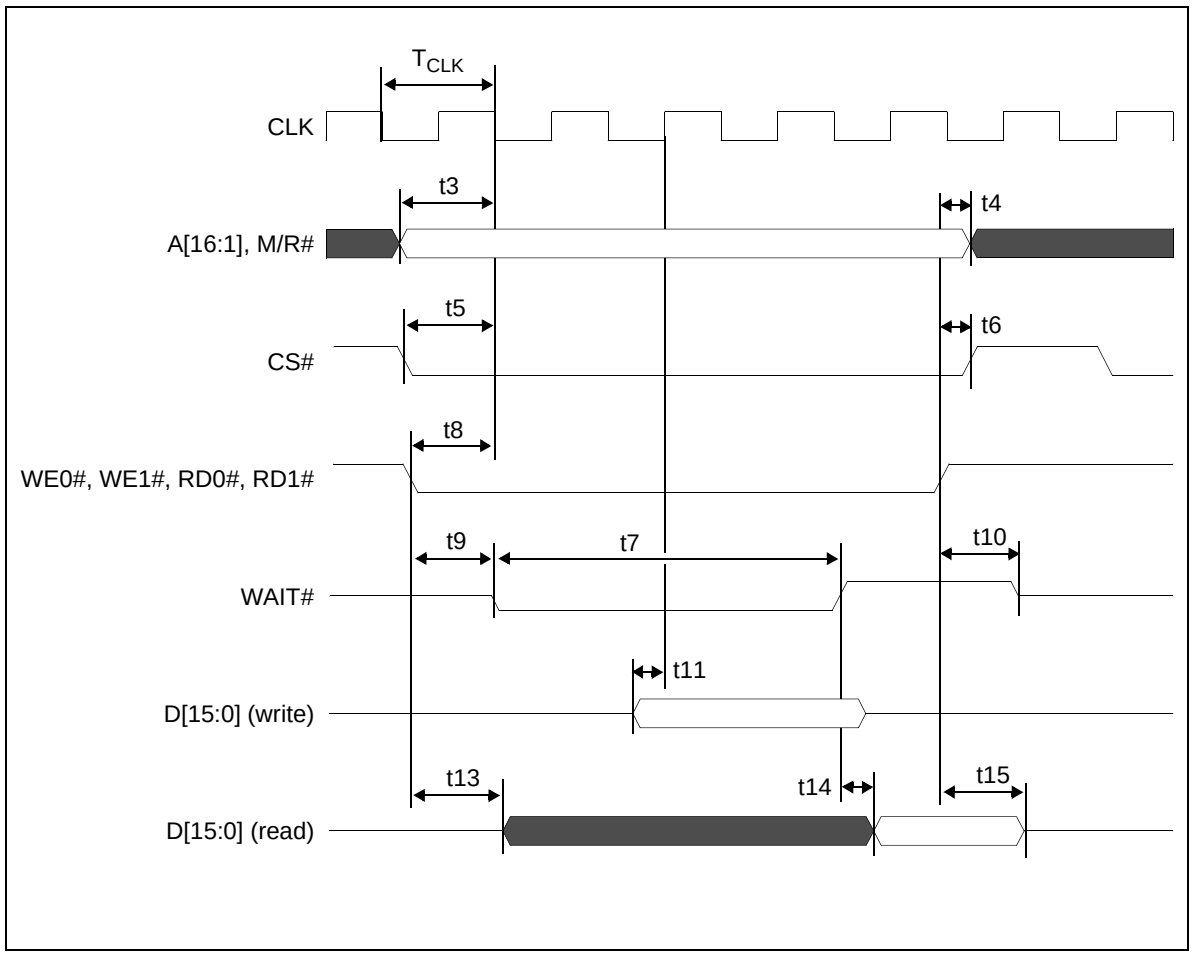


Figure 6-2 Generic #1 Interface Timing

Note
The above diagram assumes that $\overline{\text{WAIT\#}}$ is used and that MCLK = BCLK.

Table 6-5: Generic #1 Interface Timing

Symbol	Parameter	Min	Max	Unit
f_{CLK}	Bus Clock frequency		50	MHz
T_{CLK}	Bus Clock period	$1/f_{CLK}$		ns
t3	A[16:0], M/R# setup to first CLK rising edge where CS# = 0 and either RD0#, RD1# = 0 or WE0#, WE1# = 0	1		ns
t4	A[16:0], M/R# hold from either RD0#, RD1# or WE0#, WE1# rising edge	0		ns
t5	CS# setup to CLK rising edge	1		ns
t6	CS# hold from either RD0#, RD1# or WE0#, WE1# rising edge	0		ns
t7a	WAIT# asserted for MCLK = BCLK		8	T_{CLK}
t7b	WAIT# asserted for MCLK = BCLK ÷ 2		13	T_{CLK}
t7c	WAIT# asserted for MCLK = BCLK ÷ 3		17	T_{CLK}
t7d	WAIT# asserted for MCLK = BCLK ÷ 4		20	T_{CLK}
t8	RD0#, RD1#, WE0#, WE1# setup to CLK rising edge	1		ns
t9	Falling edge of either RD0#, RD1# or WE0#, WE1# to WAIT# driven low	5	12	ns
t10	Rising edge of either RD0#, RD1# or WE0#, WE1# to WAIT# high impedance	3	8	ns
t11	D[15:0] setup to third CLK rising edge where CS# = 0 and WE0#, WE1# = 0 (write cycle) (see note 1)	1		ns
t12	D[15:0] hold from WAIT# rising edge (write cycle)	0		ns
t13	RD0#, RD1# falling edge to D[15:0] driven (read cycle)	4	11	ns
t14	WAIT# rising edge to D[15:0] valid (read cycle)		2	ns
t15	RD0#, RD1# rising edge to D[15:0] high impedance (read cycle)	3	9	ns

1. t11 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.2.2 Generic #2 Interface Timing

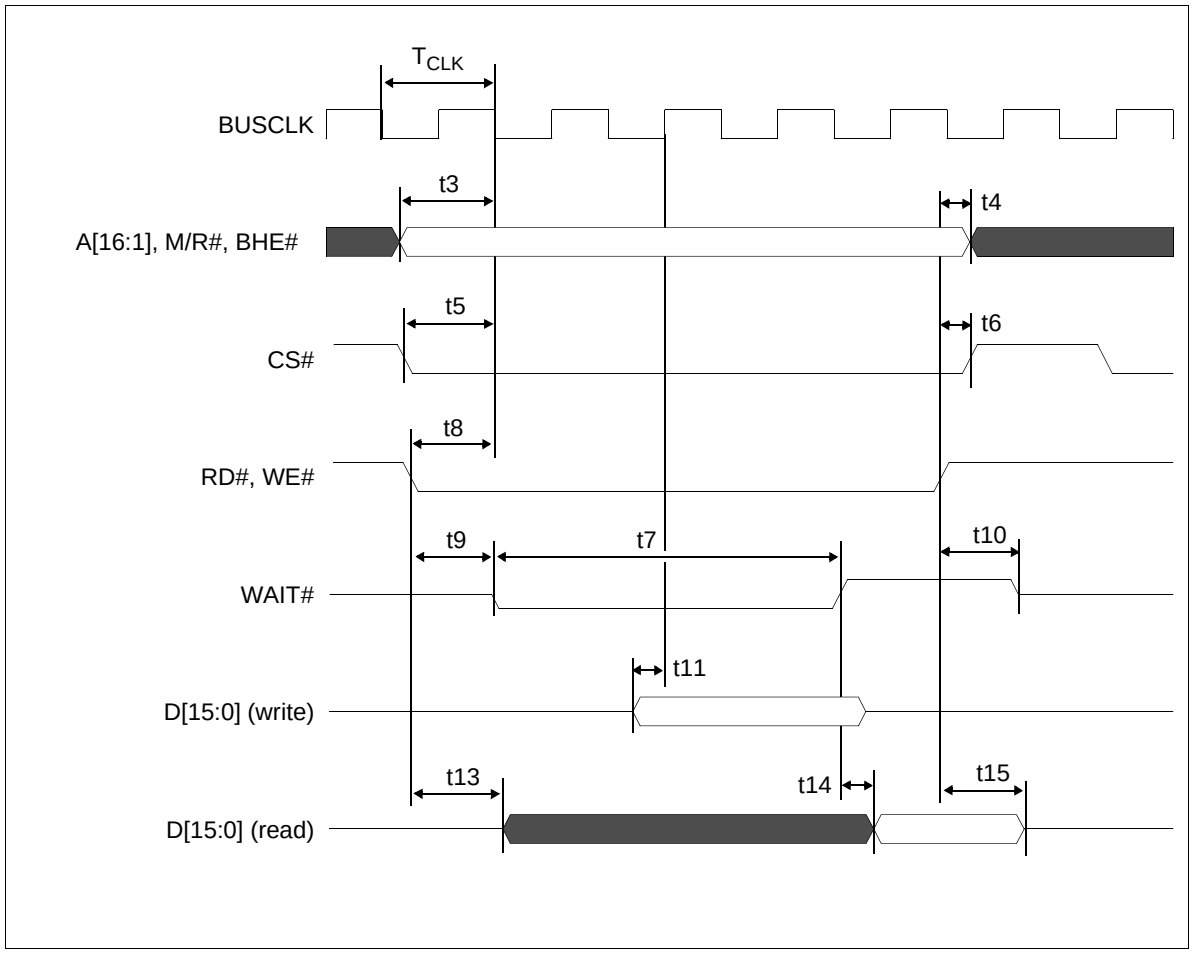


Figure 6-3 Generic #2 Interface Timing

Table 6-6: Generic #2 Interface Timing

Symbol	Parameter	Min	Max	Unit
f_{BUSCLK}	Bus Clock frequency		50	MHz
T_{BUSCLK}	Bus Clock period	$1/f_{BUSCLK}$		ns
t3	A[16:0], M/R#, BHE# setup to first BUSCLK rising edge where CS# = 0 and either RD# = 0 or WE# = 0	1		ns
t4	A[16:0], M/R#, BHE# hold from either RD# or WE# rising edge	0		ns
t5	CS# setup to BUSCLK rising edge	1		ns
t6	CS# hold from either RD# or WE# rising edge	0		ns
t7a	WAIT# asserted for MCLK = BCLK		8	T_{BUSCLK}
t7b	WAIT# asserted for MCLK = BCLK ÷ 2		13	T_{BUSCLK}
t7c	WAIT# asserted for MCLK = BCLK ÷ 3		15	T_{BUSCLK}
t7d	WAIT# asserted for MCLK = BCLK ÷ 4		21	T_{BUSCLK}
t8	RD# or WE# setup to BUSCLK rising edge	1		ns
t9	Falling edge of either RD# or WE# to WAIT# driven low	5	12	ns
t10	Rising edge of either RD# or WE# to WAIT# high impedance	3	8	ns
t11	D[15:0] setup to third BUSCLK rising edge where CS# = 0 and WE# = 0 (write cycle) (see note 1)	1		ns
t12	D[15:0] hold from WAIT# rising edge (write cycle)	0		ns
t13	RD# falling edge to D[15:0] driven (read cycle)	5	11	ns
t14	WAIT# rising edge to D[15:0] valid (read cycle)		2	ns
t15	Rising edge of RD# to D[15:0] high impedance (read cycle)	3	9	ns

1. t11 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.2.3 Hitachi SH-4 Interface Timing

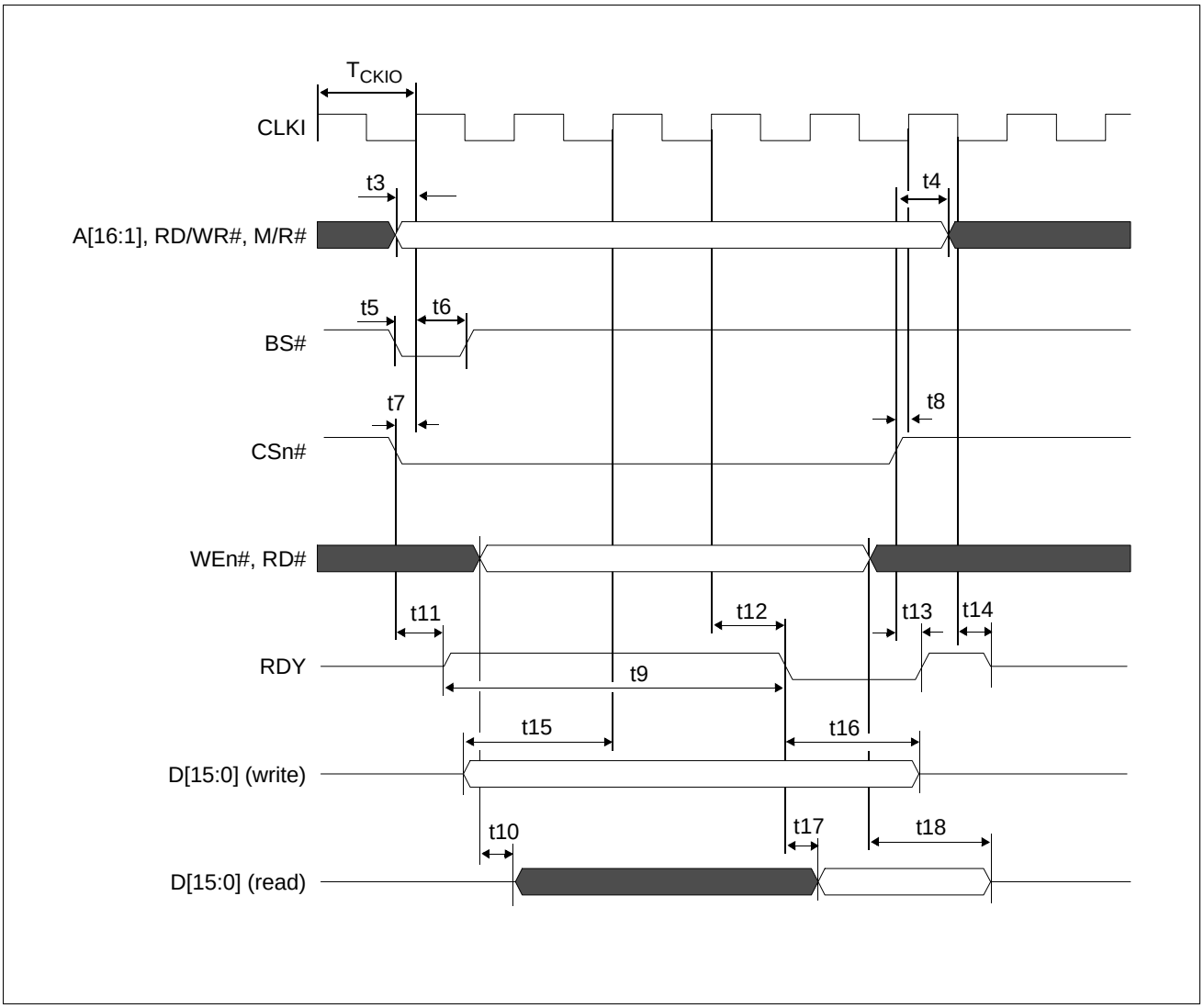


Figure 6-4 Hitachi SH-4 Interface Timing

Table 6-7: Hitachi SH-4 Interface Timing

Symbol	Parameter	Min	Max	Unit
f_{CKIO}	Clock frequency		66	MHz
T_{CKIO}	Clock period	$1/f_{CKIO}$		ns
t3	A[16:0], M/R#, RD/WR# setup to CKIO	1		ns
t4	A[16:0], M/R#, RD/WR# hold from CSn#	0		ns
t5	BS# setup	1		ns
t6	BS# hold	5		ns
t7	CSn# setup	1		ns
t8	CSn# high setup to CKIO	2		ns
t9a	RDY asserted for MCLK = BCLK (max. MCLK = 50MHz)		7	T_{CKIO}
t9b	RDY asserted for MCLK = BCLK ÷ 2		14	T_{CKIO}
t9c	RDY asserted for MCLK = BCLK ÷ 3		16	T_{CKIO}
t9d	RDY asserted for MCLK = BCLK ÷ 4		23	T_{CKIO}
t10	Falling edge RD# to D[15:0] driven (read cycle)	4	9	ns
t11	Falling edge CSn# to RDY# driven high	4	9	ns
t12	CKIO to RDY# low	5	14	ns
t13	CSn# high to RDY# high	5	12	ns
t14	Falling edge CKIO to RDY# high impedance	4	10	ns
t15	D[15:0] setup to 2nd CKIO after BS# (write cycle) (see note 1)	0		ns
t16	D[15:0] hold (write cycle)	0		ns
t17	RDY# falling edge to D[15:0] valid (read cycle)		2	ns
t18	Rising edge RD# to D[15:0] high impedance (read cycle)	3	9	ns

1. t15 is the delay from when data is placed on the bus until the data is latched into the write buffer.

Note
Minimum one software WAIT state is required.

6.2.4 Hitachi SH-3 Interface Timing

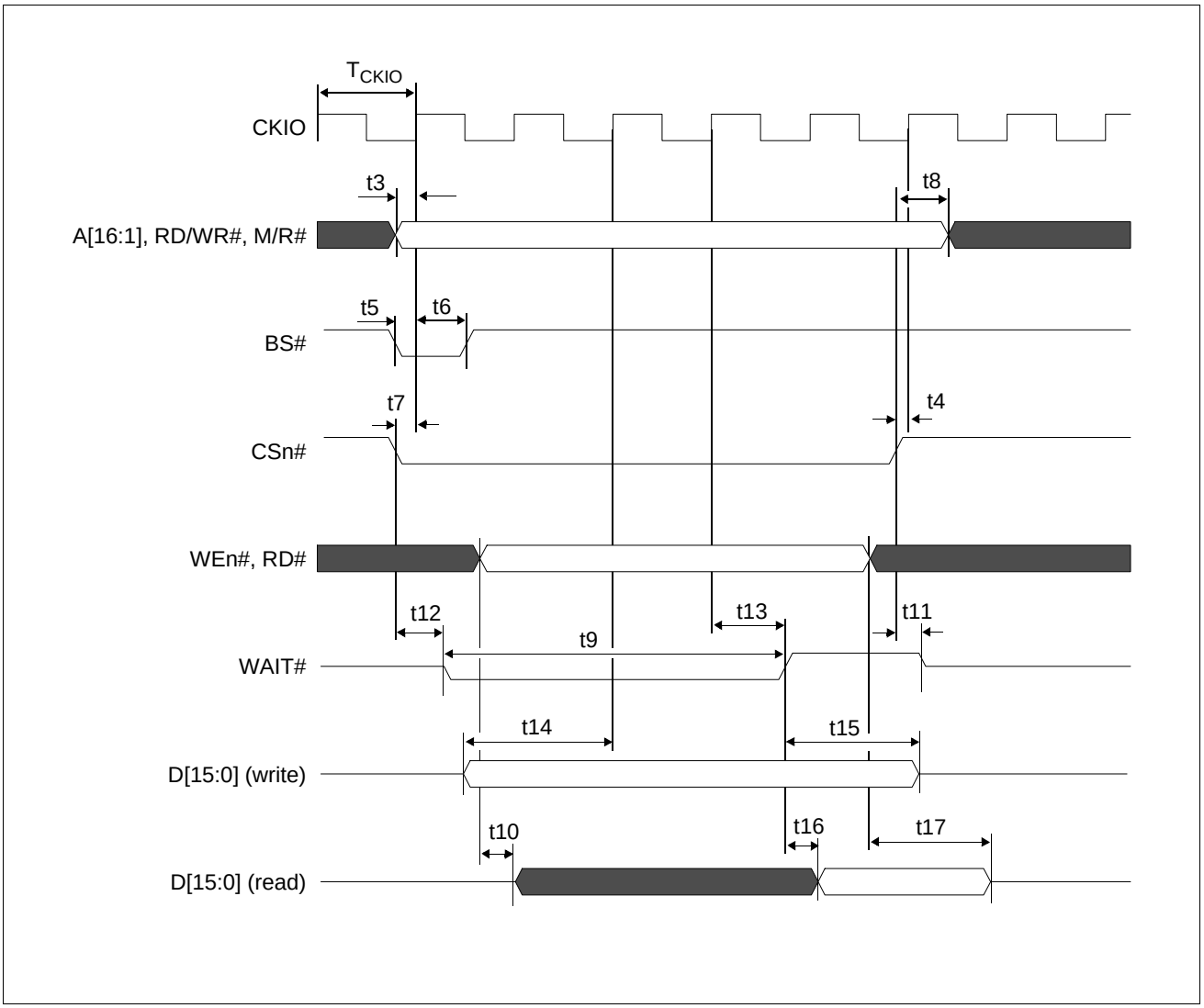


Figure 6-5 Hitachi SH-3 Interface Timing

Table 6-8: Hitachi SH-3 Interface Timing

Symbol	Parameter	Min	Max	Unit
f_{CKIO}	Bus Clock frequency		66	MHz
T_{CKIO}	Bus Clock period	$1/f_{CKIO}$		ns
t3	A[16:0], M/R#, RD/WR# setup to CKIO	1		ns
t4	CSn# high setup to CKIO	1		ns
t5	BS# setup	0		ns
t6	BS# hold	5		ns
t7	CSn# setup	1		ns
t8	A[16:0], M/R#, RD/WR# hold from CS#	0		ns
t9a	WAIT# asserted for MCLK = BCLK (max. MCLK = 50MHz)		6	T_{CKIO}
t9b	WAIT# asserted for MCLK = BCLK ÷ 2		13	T_{CKIO}
t9c	WAIT# asserted for MCLK = BCLK ÷ 3		15	T_{CKIO}
t9d	WAIT# asserted for MCLK = BCLK ÷ 4		23	T_{CKIO}
t10	Falling edge RD# to D[15:0] driven (read cycle)	4	9	ns
t11	Rising edge CSn# to WAIT# high impedance	3	7	ns
t12	Falling edge CSn# to WAIT# driven low	4	11	ns
t13	CKIO to WAIT# delay	5	14	ns
t14	D[15:0] setup to 2nd CKIO after BS# (write cycle) (see note 1)	0		ns
t15	D[15:0] hold (write cycle)	0		ns
t16	WAIT# rising edge to D[15:0] valid (read cycle)		2	ns
t17	Rising edge RD# to D[15:0] high impedance (read cycle)	3	9	ns

1. t14 is the delay from when data is placed on the bus until the data is latched into the write buffer.

Note
Minimum one software WAIT state is required.

6.2.5 Motorola MC68K #1 Interface Timing (e.g. MC68000)

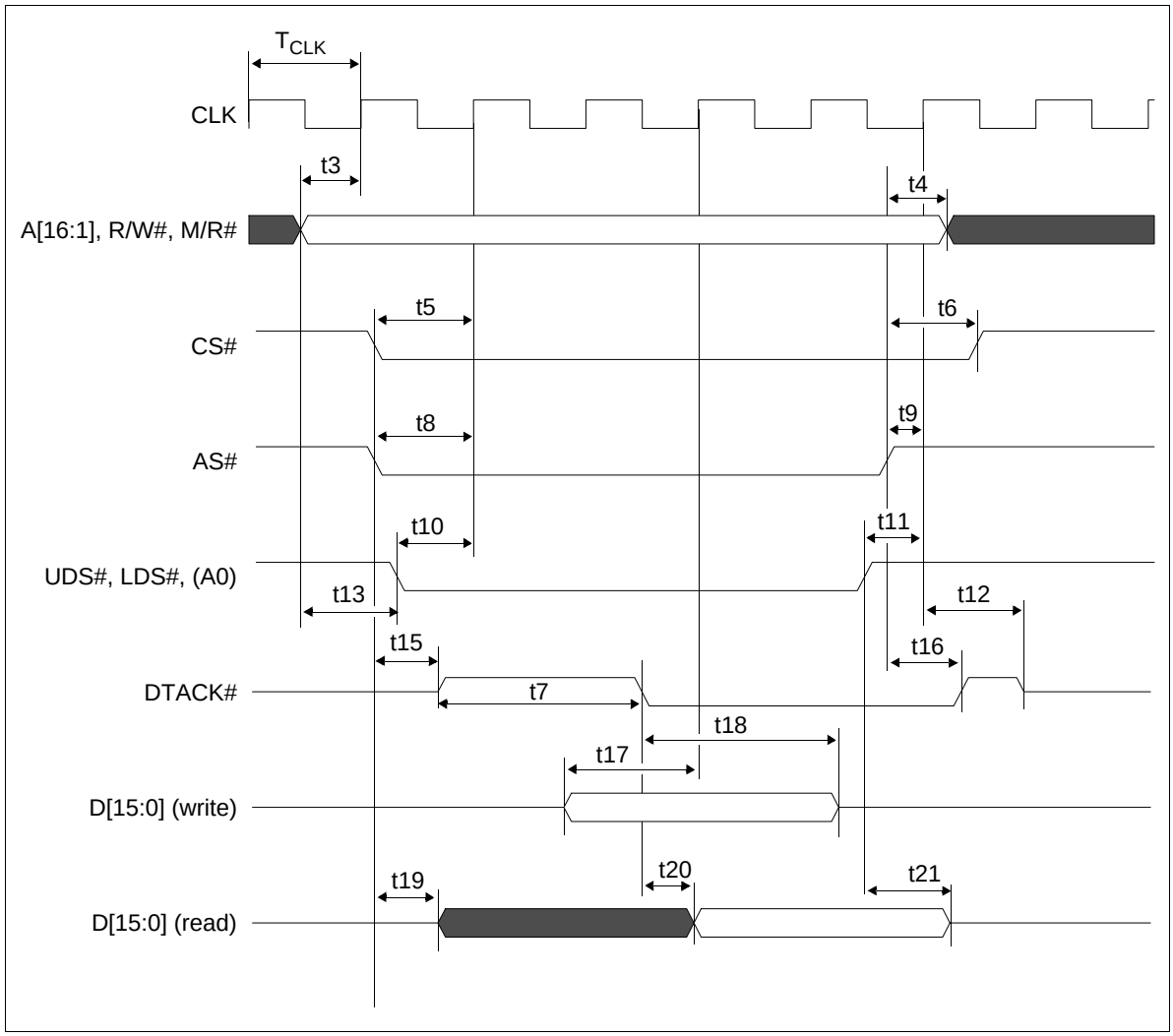


Figure 6-6 Motorola MC68K #1 Interface Timing

Table 6-9: Motorola MC68K #1 Interface Timing

Symbol	Parameter	Min	Max	Unit
f_{CLK}	Bus Clock Frequency		50	MHz
T_{CLK}	Bus Clock period	$1/f_{CLK}$		ns
t3	A[16:1], R/W#, M/R# setup to first CLK rising edge where CS# = 0, AS# = 0, UDS# = 0, and LDS# = 0	1		ns
t4	A[16:1], M/R# hold from AS# rising edge	0		ns
t5	CS# setup to CLK rising edge while CS#, AS#, UDS#/LDS# = 0	1		ns
t6	CS# hold from AS# rising edge	0		ns
t7a	DTACK# asserted for MCLK = BCLK		8	T_{CLK}
t7b	DTACK# asserted for MCLK = BCLK ÷ 2		13	T_{CLK}
t7c	DTACK# asserted for MCLK = BCLK ÷ 3		17	T_{CLK}
t7d	DTACK# asserted for MCLK = BCLK ÷ 4		23	T_{CLK}
t8	AS# setup to CLK rising edge while CS#, AS#, UDS#/LDS# = 0	1		ns
t9	AS# setup to CLK rising edge	1		ns
t10	UDS#/LDS# setup to CLK rising edge while CS#, AS#, UDS#/LDS# = 0	1		ns
t11	UDS#/LDS# high setup to CLK rising edge	1		ns
t12	First CLK rising edge where AS# = 1 to DTACK# high impedance	3	10	ns
t13	R/W# setup to CLK rising edge before all CS#, AS#, UDS# and/or LDS# = 0	1		ns
t15	AS# = 0 and CS# = 0 to DTACK# driven high	4	10	ns
t16	AS# rising edge to DTACK# rising edge	5	14	ns
t17	D[15:0] valid to third CLK rising edge where CS# = 0, AS# = 0 and either UDS# = 0 or LDS# = 0 (write cycle) (see note 1)	1		ns
t18	D[15:0] hold from DTACK# falling edge (write cycle)	0		ns
t19	UDS# = 0 and/or LDS# = 0 to D[15:0] driven (read cycle)	3	9	ns
t20	DTACK# falling edge to D[15:0] valid (read cycle)		0	ns
t21	UDS#, LDS# rising edge to D[15:0] high impedance (read cycle)	3	4	ns

1. t17 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.2.6 Motorola MC68K #2 Interface Timing (e.g. MC68030)

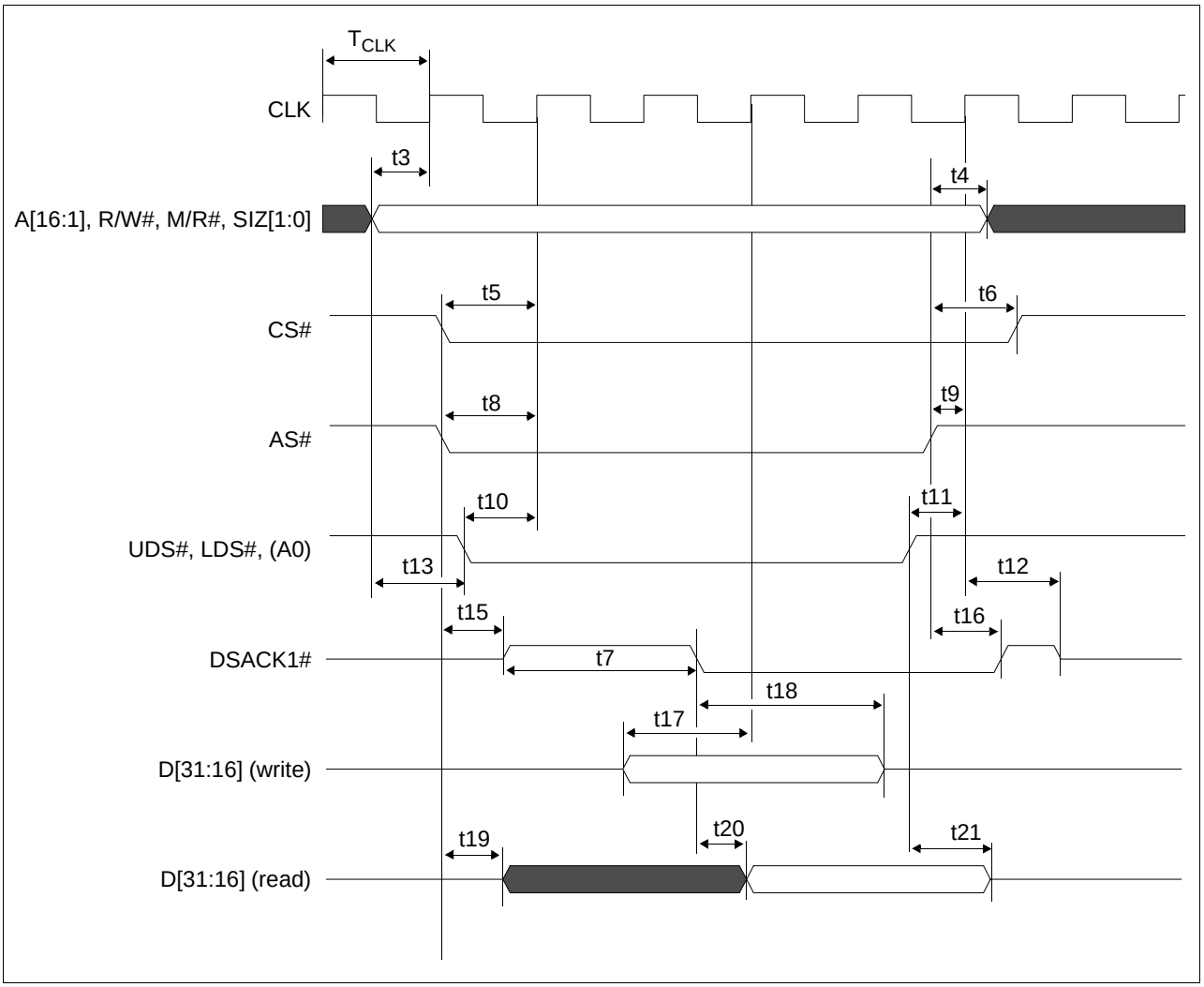


Figure 6-7 Motorola MC68K #2 Interface Timing

Table 6-10: Motorola MC68K #2 Interface Timing

Symbol	Parameter	Min	Max	Unit
f_{CLK}	Bus Clock frequency		50	MHz
T_{CLK}	Bus Clock period	$1/f_{CLK}$		ns
t3	A[16:0], SIZ[1:0], M/R# setup to first CLK rising edge where CS# = 0, AS# = 0, DS# = 0	4		ns
t4	A[16:0], SIZ[1:0], M/R#, R/W# hold from AS# rising edge	0		ns
t5	CS# setup to CLK rising edge	3		ns
t6	CS# hold from AS# rising edge	0		ns
t7a	DSACK1# asserted for MCLK = BCLK		8	T_{CLK}
t7b	DSACK1# asserted for MCLK = BCLK ÷ 2		13	T_{CLK}
t7c	DSACK1# asserted for MCLK = BCLK ÷ 3		17	T_{CLK}
t7d	DSACK1# asserted for MCLK = BCLK ÷ 4		22	T_{CLK}
t8	AS# falling edge to CLK rising edge	4		ns
t9	AS# rising edge to CLK rising edge	4		ns
t10	DS# falling edge to CLK rising edge	4		ns
t11	DS# setup to CLK rising edge	4		ns
t12	First CLK where AS# = 1 to DSACK1# high impedance	2	28	ns
t13	R/W# setup to CLK rising edge before all CS# = 0, AS# = 0, and DS# = 0	1		ns
t15	AS# = 0 and CS# = 0 to DSACK1# rising edge	3	10	ns
t16	AS# rising edge to DSACK1# rising edge	5	14	ns
t17	D[31:16] valid to third CLK rising edge where CS# = 0, AS# = 0, and DS# = 0 (write cycle) (see note 1)	1		ns
t18	D[31:16] hold from falling edge of DSACK1# (write cycle)	0		ns
t19	DS# falling edge to D[31:16] driven (read cycle)	5	14	ns
t20	DSACK1# falling edge to D[31:16] valid (read cycle)		0	ns
t21	DS# rising edge to D[31:16] invalid/high impedance (read cycle)	3	10	ns

1. t17 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.2.7 Motorola REDCAP2 Interface Timing

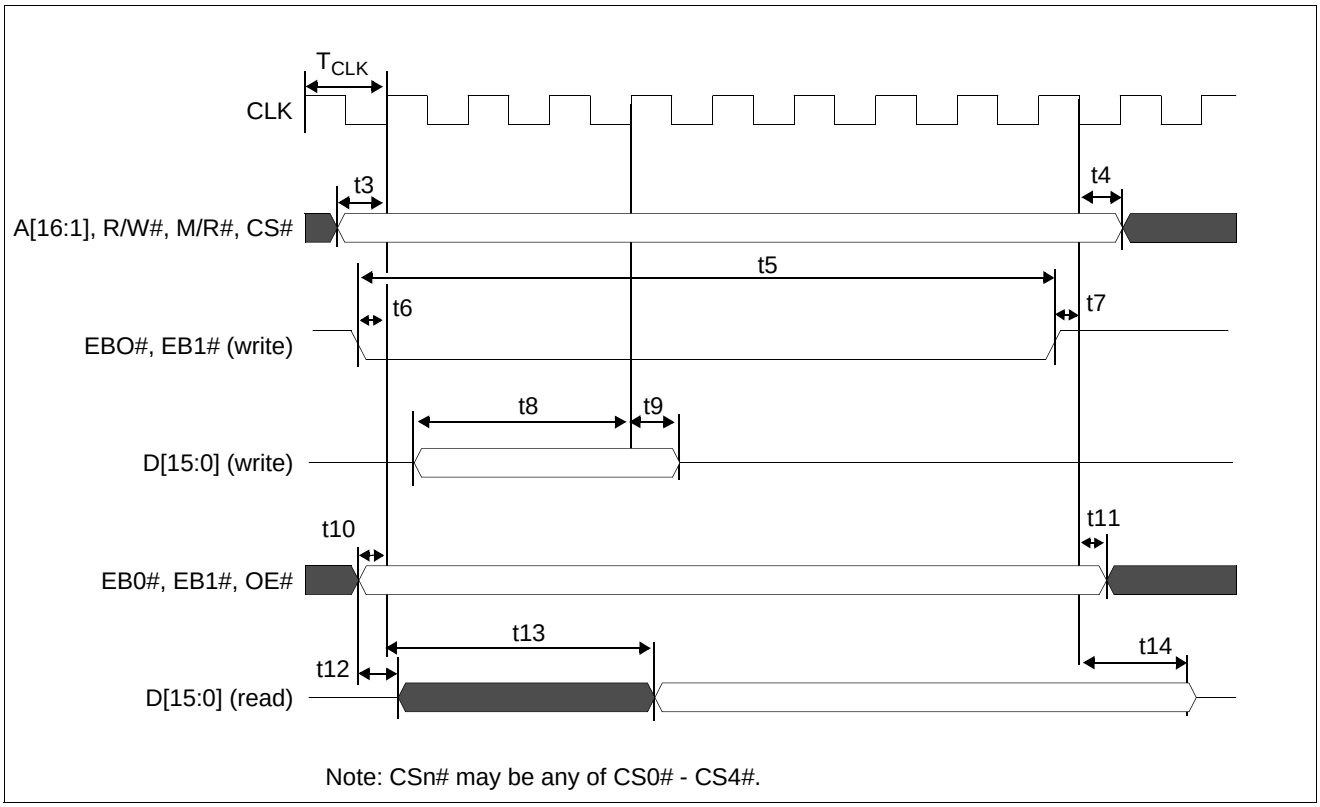


Figure 6-8 Motorola REDCAP2 Interface Timing

Table 6-11: Motorola REDCAP2 Interface Timing

Symbol	Parameter	Min	Max	Units
f_{CLK}	Bus Clock frequency		17	MHz
T_{CLK}	Bus Clock period	$1/f_{CLK}$		ns
t3	A[16:0], M/R#, R/$\overline{W}$, $\overline{CSn}$ setup to CLK rising edge	1		ns
t4	A[16:0], M/R#, R/$\overline{W}$, $\overline{CSn}$ hold from CLK rising edge	0		ns
t5a	$\overline{CSn}$ asserted for MCLK = BCLK	8		T_{CLK}
t5b	$\overline{CSn}$ asserted for MCLK = BCLK ÷ 2	10		T_{CLK}
t5c	$\overline{CSn}$ asserted for MCLK = BCLK ÷ 3	13		T_{CLK}
t5d	$\overline{CSn}$ asserted for MCLK = BCLK ÷ 4	15		T_{CLK}
t6	$\overline{EB0}$, $\overline{EB1}$ asserted to CLK rising edge (write cycle)	2		ns
t7	$\overline{EB0}$, $\overline{EB1}$ de-asserted to CLK rising edge (write cycle)	3		ns
t8	D[15:0] input setup to 3rd CLK rising edge after $\overline{EB0}$ or $\overline{EB1}$ asserted low (write cycle) (see note 1)	1		ns
t9	D[15:0] input hold from 3rd CLK rising edge after $\overline{EB0}$ or $\overline{EB1}$ asserted low (write cycle)	1		ns
t10	$\overline{OE}$, $\overline{EB0}$, $\overline{EB1}$ setup to CLK rising edge (read cycle)	1		ns
t11	$\overline{OE}$, $\overline{EB0}$, $\overline{EB1}$ hold to CLK rising edge (read cycle)	1		ns
t12	D[15:0] output delay from $\overline{OE}$, $\overline{EB0}$, $\overline{EB1}$ falling edge (read cycle)	5	11	ns
t13a	1st CLK rising edge after $\overline{EB0}$ or $\overline{EB1}$ asserted low to D[15:0] valid for MCLK = BCLK (read cycle)		5CLK + 16	ns
t13b	1st CLK rising edge after $\overline{EB0}$ or $\overline{EB1}$ asserted low to D[15:0] valid for MCLK = BCLK ÷ 2 (read cycle)		8CLK + 16	ns
t13c	1st CLK rising edge after $\overline{EB0}$ or $\overline{EB1}$ asserted low to D[15:0] valid for MCLK = BCLK ÷ 3 (read cycle)		9CLK + 16	ns
t13d	1st CLK rising edge after $\overline{EB0}$ or $\overline{EB1}$ asserted low to D[15:0] valid for MCLK = BCLK ÷ 4 (read cycle)		11CLK + 16	ns
t14	CLK rising edge to D[15:0] output in Hi-Z (read cycle)	3	9	ns

1. t8 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.2.8 Motorola DragonBall Interface Timing with $\overline{\text{DTACK}}$ (e.g. MC68EZ328/MC68VZ328)

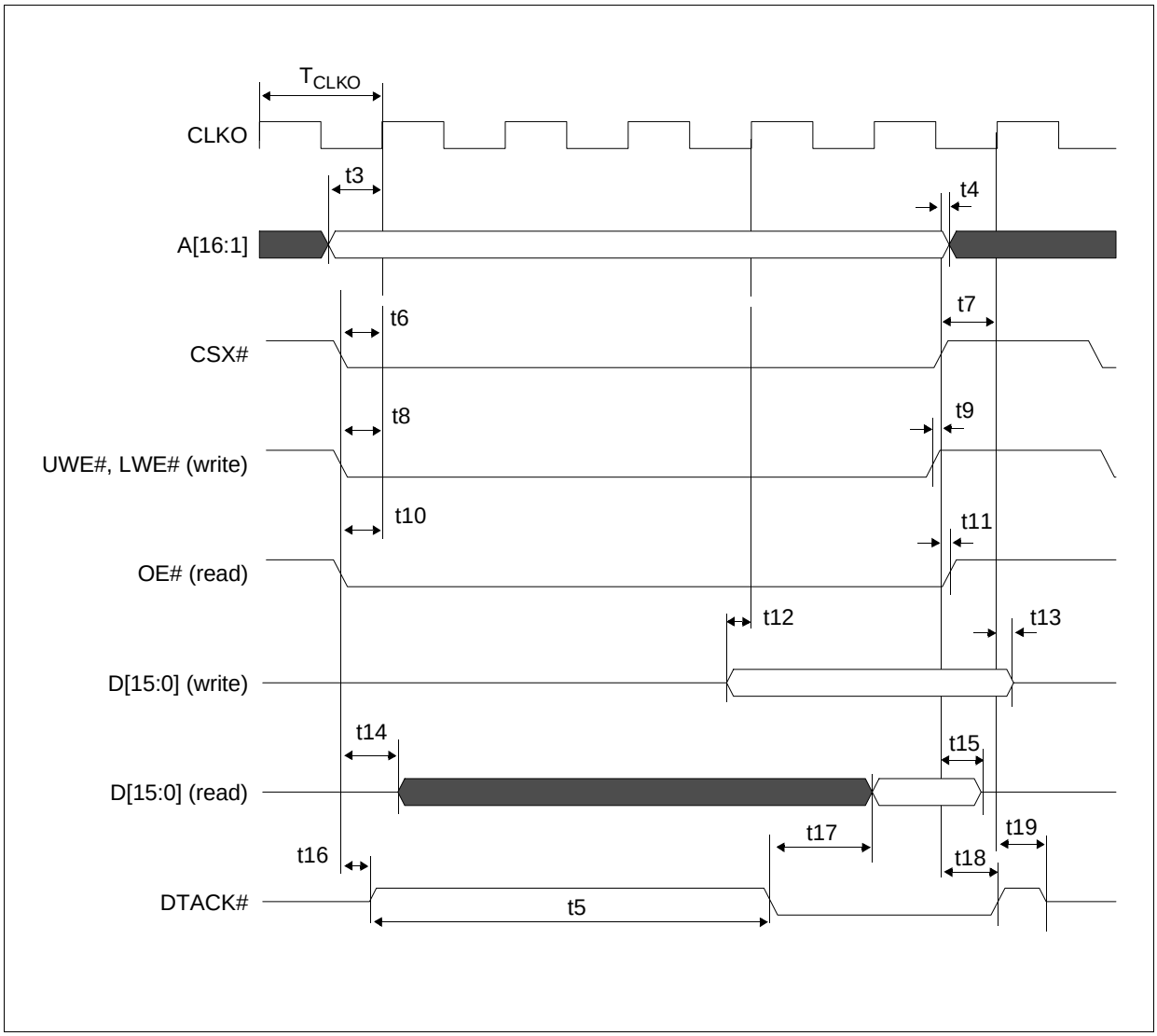


Figure 6-9 Motorola DragonBall Interface with $\overline{\text{DTACK}}$ Timing

Table 6-12: Motorola DragonBall Interface with $\overline{DTACK}$ Timing

Symbol	Parameter	MC68EZ328		MC68VZ328		Unit
		Min	Max	Min	Max	
f_{CLKO}	Bus Clock frequency		16		33	MHz
T_{CLKO}	Bus Clock period	$1/f_{CLKO}$		$1/f_{CLKO}$		ns
t3	A[16:0] setup 1st CLKO when $\overline{CSX}$ = 0 and either $\overline{UWE}$/$\overline{LWE}$ or $\overline{OE}$ = 0	0		0		ns
t4	A[16:0] hold from $\overline{CSX}$ rising edge	0		0		ns
t5a	$\overline{DTACK}$ asserted for MCLK = BCLK		8		8	T_{CLKO}
t5b	$\overline{DTACK}$ asserted for MCLK = BCLK ÷ 2		13		13	T_{CLKO}
t5c	$\overline{DTACK}$ asserted for MCLK = BCLK ÷ 3		16		16	T_{CLKO}
t5d	$\overline{DTACK}$ asserted for MCLK = BCLK ÷ 4		22		22	T_{CLKO}
t6	$\overline{CSX}$ setup to CLKO rising edge	2		2		ns
t7	$\overline{CSX}$ rising edge to CLKO rising edge	2		2		ns
t8	$\overline{UWE}$/$\overline{LWE}$ falling edge to CLKO rising edge	2		2		ns
t9	$\overline{UWE}$/$\overline{LWE}$ rising edge to $\overline{CSX}$ rising edge	0		0		ns
t10	$\overline{OE}$ falling edge to CLKO rising edge	2		2		ns
t11	$\overline{OE}$ hold from $\overline{CSX}$ rising edge	0		0		ns
t12	D[15:0] setup to 3rd CLKO when $\overline{CSX}$, $\overline{UWE}$/$\overline{LWE}$ asserted (write cycle) (see note 1)	0		1		ns
t13	D[15:0] in hold from $\overline{CSX}$ rising edge (write cycle)	0		0		ns
t14	Falling edge of $\overline{OE}$ to D[15:0] driven (read cycle)	4	10	4	10	ns
t15	$\overline{CSX}$ rising edge to D[15:0] output Hi-Z (read cycle)	3	9	3	9	ns
t16	$\overline{CSX}$ falling edge to $\overline{DTACK}$ driven high	4	9	4	10	ns
t17	$\overline{DTACK}$ falling edge to D[15:0] valid (read cycle)		0		0	ns
t18	$\overline{CSX}$ high to $\overline{DTACK}$ high	5	13	5	14	ns
t19	CLKO rising edge to $\overline{DTACK}$ Hi-Z	4	9	4	10	ns

1. t12 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.2.9 Motorola DragonBall Interface Timing w/o $\overline{\text{DTACK}}$ (e.g. MC68EZ328/MC68VZ328)

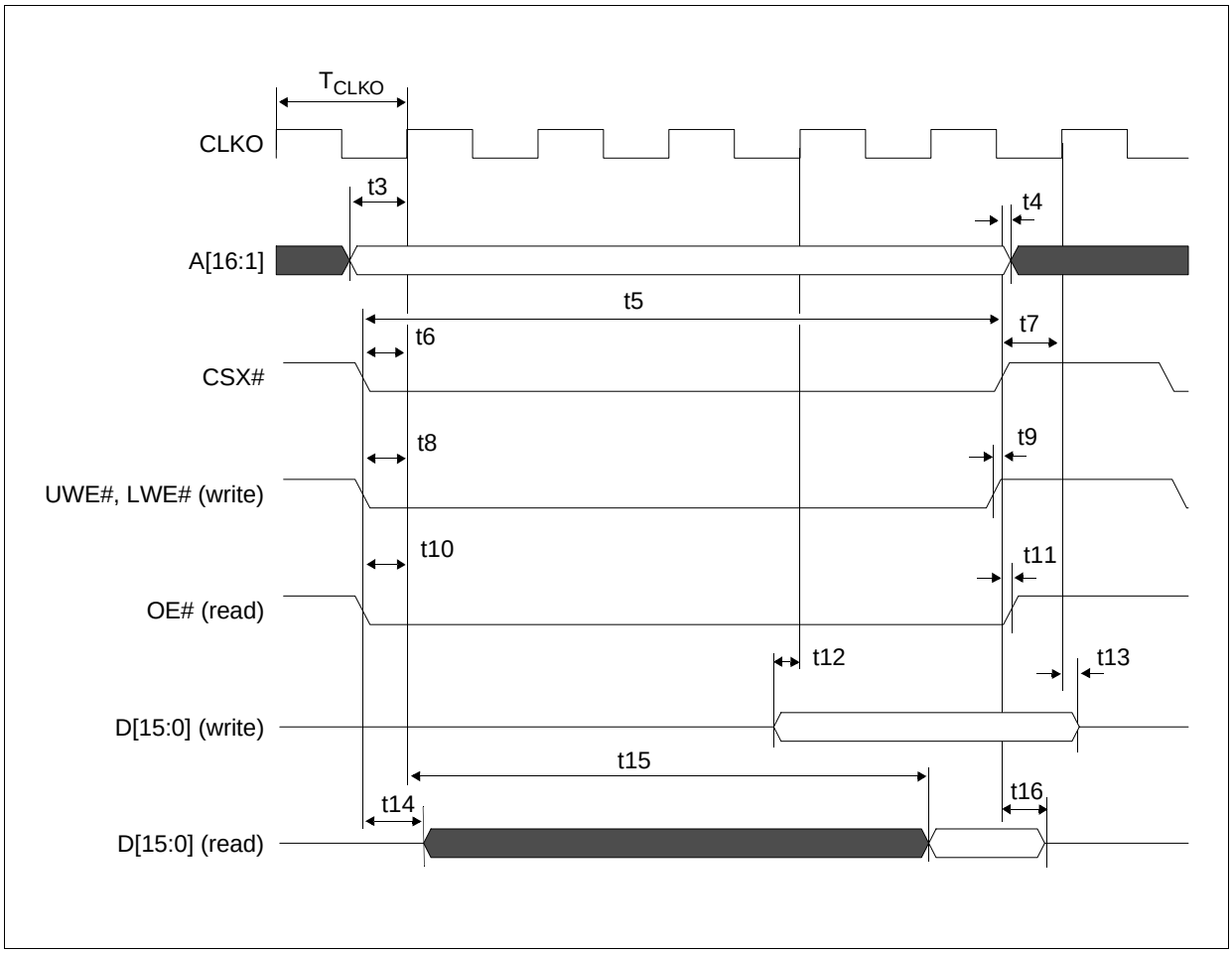


Figure 6-10 Motorola DragonBall Interface without DTACK# Timing

Table 6-13: Motorola DragonBall Interface without $\overline{DTACK}$ Timing

Symbol	Parameter	MC68EZ328		MC68VZ328		Unit
		Min	Max	Min	Max	
f_{CLKO}	Bus Clock frequency		16		33	MHz
T_{CLKO}	Bus Clock period	$1/f_{CLKO}$		$1/f_{CLKO}$		ns
t3	A[16:0] setup 1st CLKO when $\overline{CSX}$ = 0 and either $\overline{UWE}$/$\overline{LWE}$ or $\overline{OE}$ = 0	0		0		ns
t4	A[16:0] hold from $\overline{CSX}$ rising edge	0		0		ns
t5a	$\overline{CSX}$ asserted for MCLK = BCLK (CPU wait state register should be programmed to 4 wait states)		7		7	T_{CLKO}
t5b	$\overline{CSX}$ asserted for MCLK = BCLK ÷ 2 (CPU wait state register should be programmed to 8 wait states)	Not Supported			11	T_{CLKO}
t5c	$\overline{CSX}$ asserted for MCLK = BCLK ÷ 3 (CPU wait state register should be programmed to 10 wait states)	Not Supported			13	T_{CLKO}
t6	$\overline{CSX}$ setup to CLKO rising edge	2		2		ns
t7	$\overline{CSX}$ rising edge setup to CLKO rising edge	2		2		ns
t8	$\overline{UWE}$/$\overline{LWE}$ setup to CLKO rising edge	2		2		ns
t9	$\overline{UWE}$/$\overline{LWE}$ rising edge to $\overline{CSX}$ rising edge	0		0		ns
t10	$\overline{OE}$ setup to CLKO rising edge	2		2		ns
t11	$\overline{OE}$ hold from $\overline{CSX}$ rising edge	0		0		ns
t12	D[15:0] setup to 3rd CLKO after $\overline{CSX}$, $\overline{UWE}$/$\overline{LWE}$ asserted (write cycle) (see note 1)	1		1		ns
t13	$\overline{CSX}$ rising edge to D[15:0] output Hi-Z (write cycle)	0		0		ns
t14	Falling edge of $\overline{OE}$ to D[15:0] driven (read cycle)	5	11	5	11	ns
t15a	1st CLKO rising edge after $\overline{OE}$ and $\overline{CSX}$ asserted low to D[15:0] valid for MCLK = BCLK (read cycle)		$6T_{CLKO} + 6$		$6T_{CLKO} + 6$	ns
t15b	1st CLKO rising edge after $\overline{OE}$ and $\overline{CSX}$ asserted low to D[15:0] valid for MCLK = BCLK ÷ 2 (read cycle)	Not Supported			$9T_{CLKO} + 6$	ns
t15c	1st CLKO rising edge after $\overline{OE}$ and $\overline{CSX}$ asserted low to D[15:0] valid for MCLK = BCLK ÷ 3 (read cycle)	Not Supported			$12T_{CLKO} + 6$	ns
t16	$\overline{CSX}$ rising edge to D[15:0] output Hi-Z (read cycle)	3	9	3	9	ns

1. t12 is the delay from when data is placed on the bus until the data is latched into the write buffer.

6.2.10 Indirect Interface Timing (Mode 68)

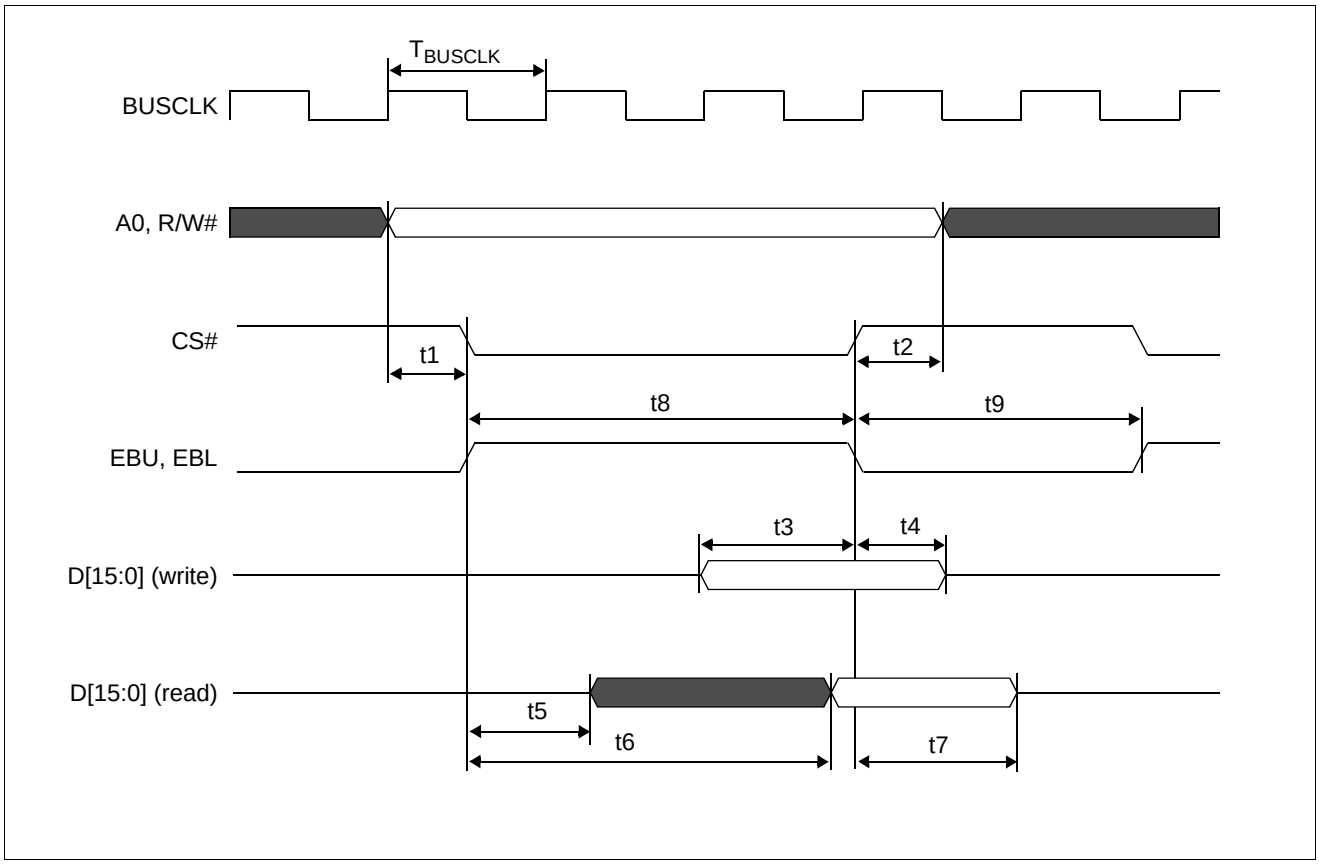


Figure 6-11 Indirect Interface Timing (Mode 68)

Table 6-14: Indirect Interface Timing (Mode 68)

Symbol	Parameter	Min	Max	Units
f_{BUSCLK}	Bus Clock frequency		50	MHz
T_{BUSCLK}	Bus Clock period	$1/f_{BUSCLK}$		ns
t1	A0 / R/W# setup to (CS# | $\overline{EBU}$), (CS# | $\overline{EBL}$) falling edge	1		ns
t2	A0 / R/W# hold to (CS# | $\overline{EBU}$), (CS# | $\overline{EBL}$) rising edge	3		ns
t3	D[15:0] setup to (CS# | $\overline{EBU}$) rising edge (write cycle)	1		T_{BUSCLK}
t4	D[15:0] hold to (CS# | $\overline{EBU}$) rising edge (write cycle)	4		ns
t5	Falling edge of (CS# | $\overline{EBL}$) to D[15:0] driven (read cycle)	2		T_{BUSCLK}
t6a	Falling edge of (CS# | $\overline{EBL}$) to valid D[15:0] driven for MCLK = BCLK (read cycle)	7.5		T_{BUSCLK}
t6b	Falling edge of (CS# | $\overline{EBL}$) to valid D[15:0] driven for MCLK = BCLK/2 (read cycle)	9.5		T_{BUSCLK}
t6c	Falling edge of (CS# | $\overline{EBL}$) to valid D[15:0] driven for MCLK = BCLK/3 (read cycle)	12.5		T_{BUSCLK}
t6d	Falling edge of (CS# | $\overline{EBL}$) to valid D[15:0] driven for MCLK = BCLK/4 (read cycle)	16.5		T_{BUSCLK}
t7	Valid D[15:0] hold to (CS# | $\overline{EBL}$) rising edge	0.5		T_{BUSCLK}
t8a	(CS# | $\overline{EBU}$) High pulse width (write cycle)	4		T_{BUSCLK}
t8b	(CS# | $\overline{EBL}$) High pulse width for MCLK = BCLK (read cycle)	8		T_{BUSCLK}
t8c	(CS# | $\overline{EBL}$) High pulse width for MCLK = BCLK/2 (read cycle)	11		T_{BUSCLK}
t8d	(CS# | $\overline{EBL}$) High pulse width for MCLK = BCLK/3 (read cycle)	15		T_{BUSCLK}
t8e	(CS# | $\overline{EBL}$) High pulse width for MCLK = BCLK/4 (read cycle)	17		T_{BUSCLK}
t9a	(CS# | $\overline{EBL}$) Low pulse width (read turnaround)	2.5		T_{BUSCLK}
t9b	(CS# | $\overline{EBU}$) Low pulse width for MCLK = BCLK (write turnaround)	2.5		T_{BUSCLK}
t9c	(CS# | $\overline{EBU}$) Low pulse width for MCLK = BCLK/2 (write turnaround)	5.5		T_{BUSCLK}
t9d	(CS# | $\overline{EBU}$) Low pulse width for MCLK = BCLK/3 (write turnaround)	7.5		T_{BUSCLK}
t9e	(CS# | $\overline{EBU}$) Low pulse width for MCLK = BCLK/4 (write turnaround)	9.5		T_{BUSCLK}

Note

Max frequency (f_{BUSCLK}) when using crystal oscillator is 12MHz.

6.2.11 Indirect Interface Timing (Mode 80)

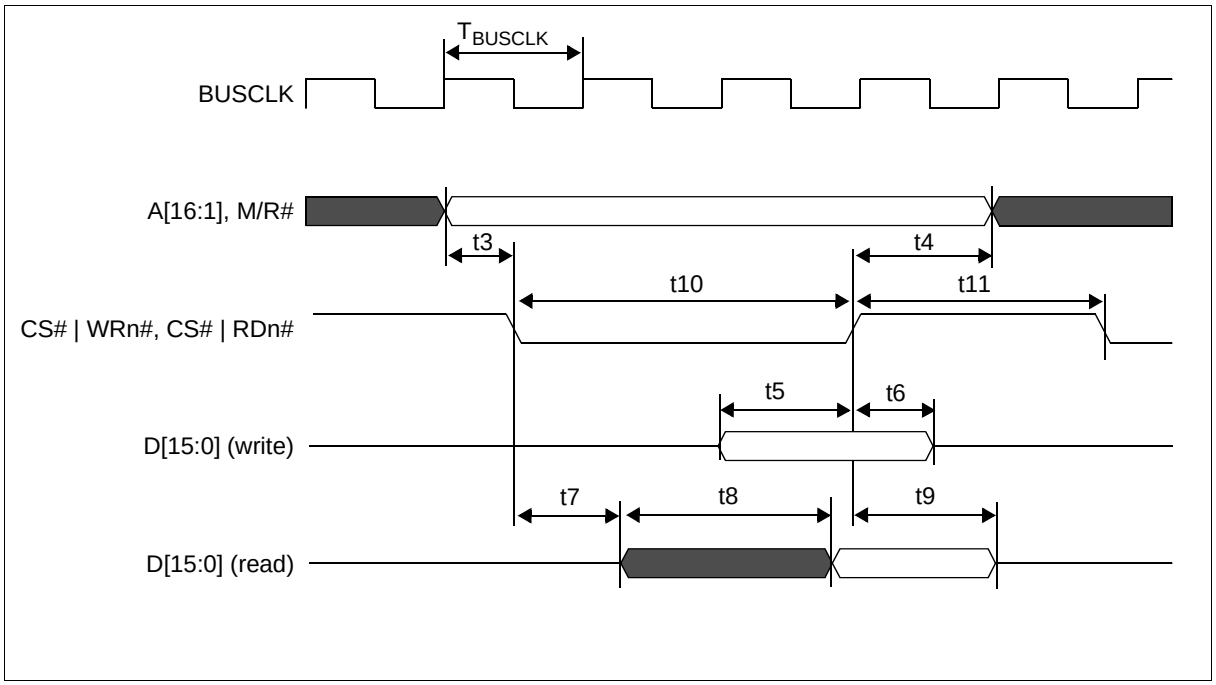


Figure 6-12 Indirect Interface Timing (Mode 80)

Table 6-15: Indirect Interface Timing (Mode 80)

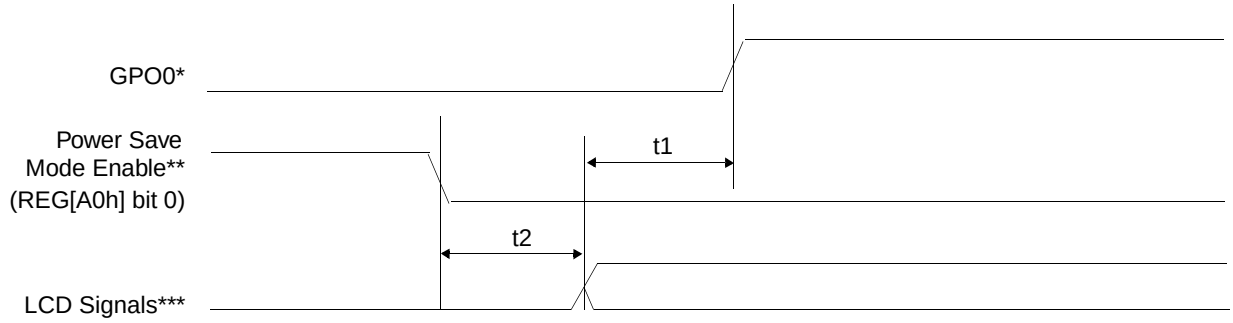
Symbol	Parameter	Min	Max	Units
f_{BUSCLK}	Bus Clock frequency		50	MHz
T_{BUSCLK}	Bus Clock period	$1/f_{BUSCLK}$		ns
t3	A0 setup to (CS# | WRn#), (CS# | RDn#) falling edge	1		ns
t4	A0 hold to (CS# | WRn#), (CS# | RDn#) rising edge	3		ns
t5	D[15:0] setup to (CS# | WRn#) rising edge (write cycle)	1		T_{BUSCLK}
t6	D[15:0] hold to (CS# | WRn#) rising edge (write cycle)	4		ns
t7	Falling edge of (CS# | RDn#) to D[15:0] driven (read cycle)	2		T_{BUSCLK}
t8a	Falling edge of (CS# | RDn#) to valid D[15:0] driven for MCLK = BCLK (read cycle)	7.5		T_{BUSCLK}
t8b	Falling edge of (CS# | RDn#) to valid D[15:0] driven for MCLK = BCLK/2 (read cycle)	10.5		T_{BUSCLK}
t8c	Falling edge of (CS# | RDn#) to valid D[15:0] driven for MCLK = BCLK/3 (read cycle)	13.5		T_{BUSCLK}
t8d	Falling edge of (CS# | RDn#) to valid D[15:0] driven for MCLK = BCLK/4 (read cycle)	15.5		T_{BUSCLK}
t9	Valid D[15:0] hold to (CS# | RDn#) rising edge	0.5		T_{BUSCLK}
t10a	(CS# | WRn#) Low pulse width (write cycle)	4		T_{BUSCLK}
t10b	(CS# | RDn#) Low pulse width for MCLK = BCLK (read cycle)	8		T_{BUSCLK}
t10c	(CS# | RDn#) Low pulse width for MCLK = BCLK/2 (read cycle)	11		T_{BUSCLK}
t10d	(CS# | RDn#) Low pulse width for MCLK = BCLK/3 (read cycle)	15		T_{BUSCLK}
t10e	(CS# | RDn#) Low pulse width for MCLK = BCLK/4 (read cycle)	17		T_{BUSCLK}
t11a	(CS# | RDn#) High pulse width (read turnaround)	2		T_{BUSCLK}
t11b	(CS# | WRn#) High pulse width for MCLK = BCLK (write turnaround)	2.5		T_{BUSCLK}
t11c	(CS# | WRn#) High pulse width for MCLK = BCLK/2 (write turnaround)	5.5		T_{BUSCLK}
t11d	(CS# | WRn#) High pulse width for MCLK = BCLK/3 (write turnaround)	7.5		T_{BUSCLK}
t11e	(CS# | WRn#) High pulse width for MCLK = BCLK/4 (write turnaround)	9.5		T_{BUSCLK}

Note

Max frequency (f_{BUSCLK}) when using crystal oscillator is 12MHz.

6.3 LCD Power Sequencing

6.3.1 Passive/TFT Power-On Sequence



*It is recommended to use the general purpose output pin GPO0 to control the LCD bias power.
**The LCD power-on sequence is activated by programming the Power Save Mode Enable bit (REG[A0h] bit 0) to 0.
***LCD Signals include: FPDAT[17:0], FPSHIFT, FPLINE, FPFRAME, and DRDY.

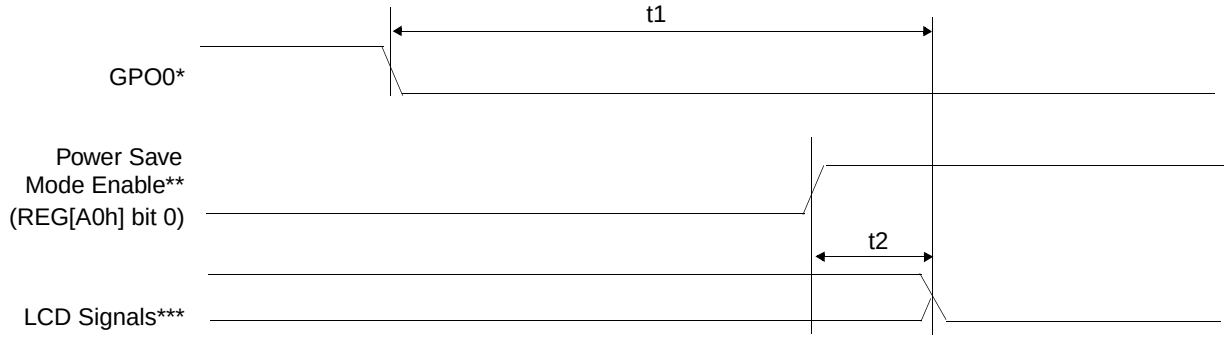
Figure 6-13 Passive/TFT Power-On Sequence Timing

Table 6-16: Passive/TFT Power-On Sequence Timing

Symbol	Parameter	Min	Max	Units
t1	LCD signals active to LCD bias active	Note 1	Note 1	
t2	Power Save Mode disabled to LCD signals active	0	20	ns

1. t1 is controlled by software and must be determined from the bias power supply delay requirements of the panel connected.

6.3.2 Passive/TFT Power-Off Sequence



*It is recommended to use the general purpose output pin GPO0 to control the LCD bias power.
**The LCD power-off sequence is activated by programming the Power Save Mode Enable bit (REG[A0h] bit 0) to 1.
***LCD Signals include: FPDAT[17:0], FPSHIFT, FPLINE, FPFRAME, and DRDY.

Figure 6-14 Passive/TFT Power-Off Sequence Timing

Table 6-17: Passive/TFT Power-Off Sequence Timing

Symbol	Parameter	Min	Max	Units
t1	LCD bias deactivated to LCD signals inactive	Note 1	Note 1	
t2	Power Save Mode enabled to LCD signals low	0	20	ns

1. t1 is controlled by software and must be determined from the bias power supply delay requirements of the panel connected.

6.4 Display Interface

The timing parameters required to drive a flat panel display are shown below. Timing details for each supported panel type are provided in the remainder of this section.

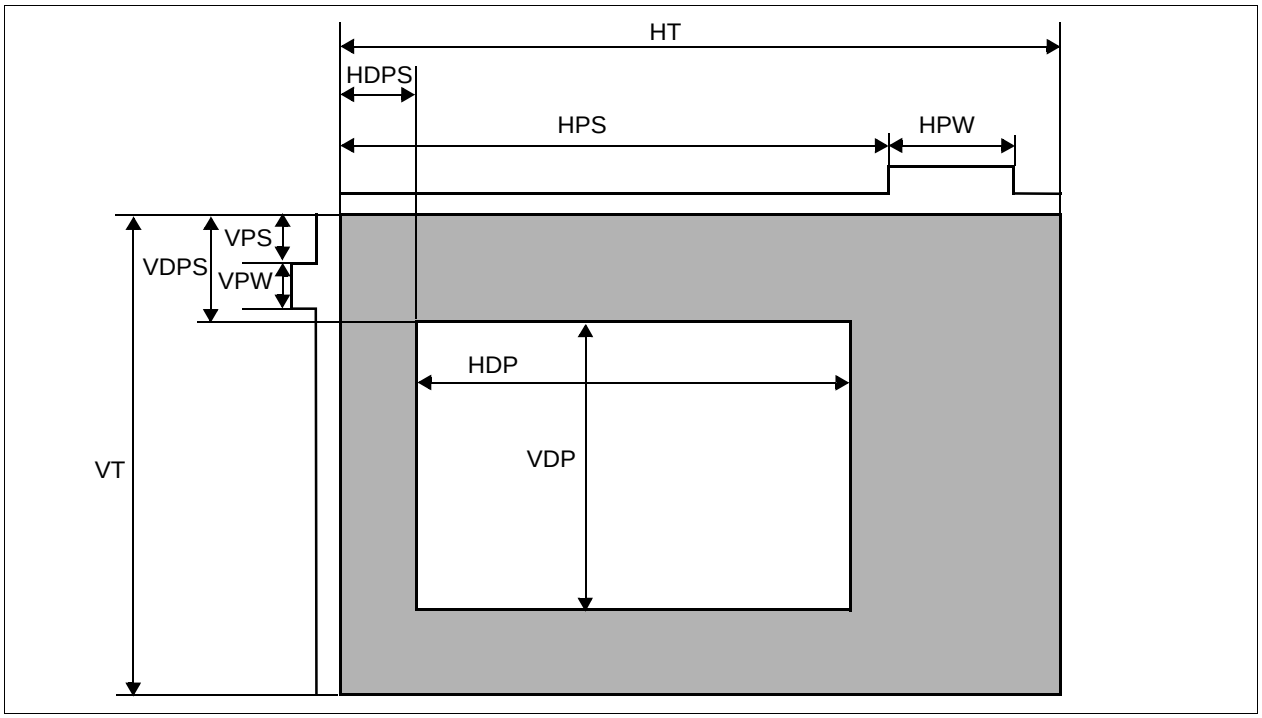


Figure 6-15 Panel Timing Parameters

Table 6-18: Panel Timing Parameter Definition and Register Summary

Symbol	Description	Derived From	Units
HT	Horizontal Total	((REG[12h] bits 6-0) + 1) x 8	Ts
HDP[1]	Horizontal Display Period[1]	((REG[44h] bits 6-0) + 1) x 8	
HDPS	Horizontal Display Period Start Position	For STN panels: ((REG[17h] bits 1-0, REG[16h] bits 7-0) + *22*)	
		For TFT panels: ((REG[17h] bits 1-0, REG[16h] bits 7-0) + *5*)	
HPS	FPLINE Pulse Start Position	(REG[23h] bits 1-0, REG[22h] bits 7-0) + 1	
HPW	FPLINE Pulse Width	(REG[20h] bits 6-0) + 1	
VT	Vertical Total	(REG[19h] bits 1-0, REG[18h] bits 7-0) + 1	Lines (HT)
VDP	Vertical Display Period	(REG[1Dh] bits 1-0, REG[1Ch] bits 7-0) + 1	
VDPS	Vertical Display Period Start Position	REG[1Fh] bits 1-0, REG[1Eh] bits 7-0	
VPS	FPFRAME Pulse Start Position	REG[27h] bits 1-0, REG[26h] bits 7-0	
VPW	FPFRAME Pulse Width	(REG[24h] bits 6-0) + 1	

1. For passive panels, the HDP must be a minimum of 32 pixels and must be increased by multiples of 16. For TFT panels, the HDP must be a minimum of 8 pixels and must be increased by multiples of 8.
2. The following formulas must be valid for all panel timings:
   HDPS + HDP < HT
   VDPS + VDP < VT

6.4.1 Generic STN Panel Timing

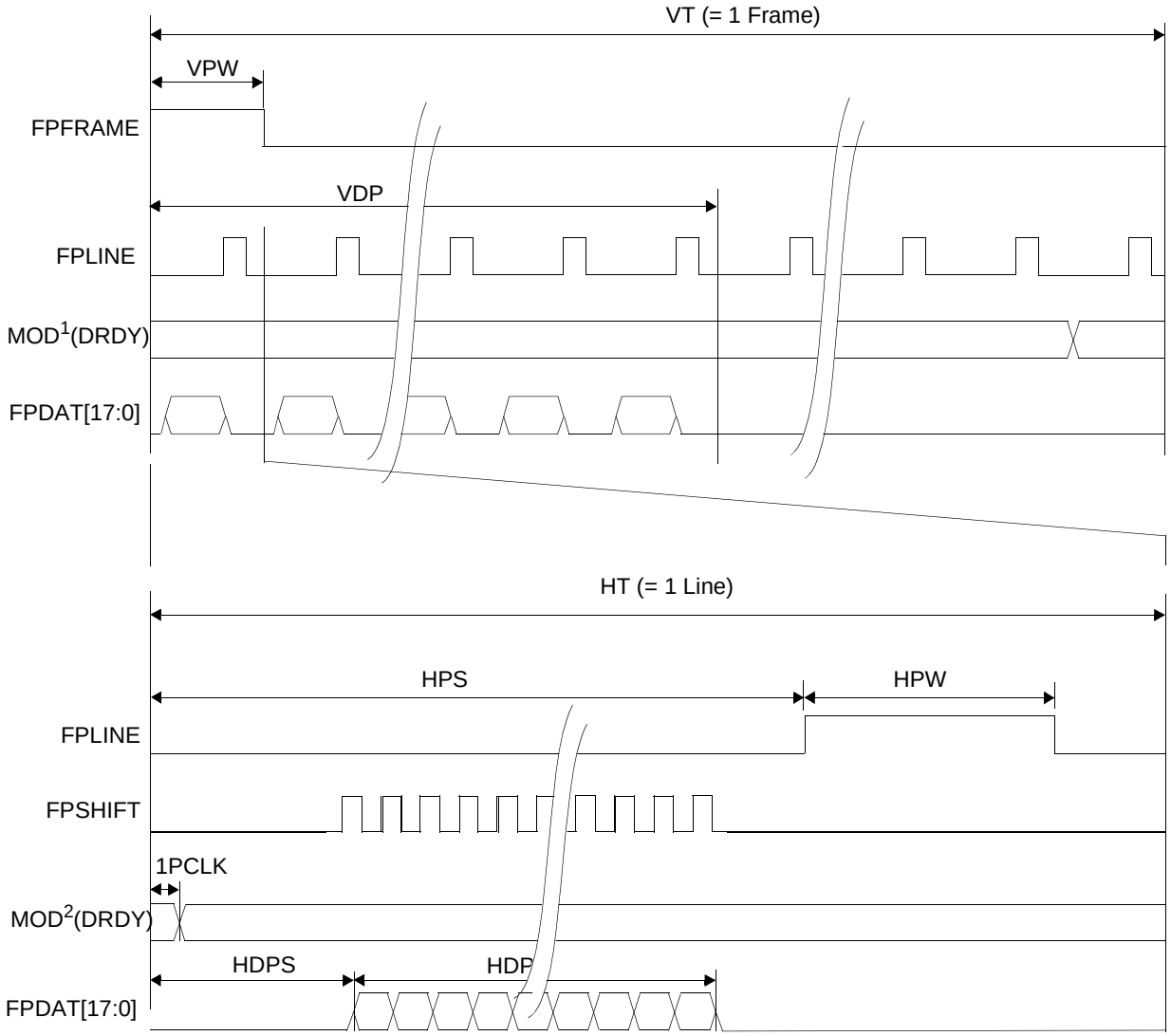


Figure 6-16 Generic STN Panel Timing

VT	= Vertical Total	= [(REG[19h] bits 1-0, REG[18h] bits 7-0) + 1] lines
VPS	= FPFRAME Pulse Start Position	= 0 lines, because [(REG[27h] bits 1-0, REG[26h] bits 7-0)] = 0
VPW	= FPFRAME Pulse Width	= [(REG[24h] bits 2-0) + 1] lines
VDPS	= Vertical Display Period Start Position	= 0 lines, because [(REG[1Fh] bits 1-0, REG[1Eh] bits 7-0)] = 0
VDP	= Vertical Display Period	= [(REG[1Dh] bits 1-0, REG[1Ch] bits 7-0) + 1] lines
HT	= Horizontal Total	= [((REG[12h] bits 6-0) + 1) x 8] pixels
HPS	= FPLINE Pulse Start Position	= [(REG[23h] bits 1-0, REG[22h] bits 7-0) + 1] pixels
HPW	= FPLINE Pulse Width	= [(REG[20h] bits 6-0) + 1] pixels
HDPS	= Horizontal Display Period Start Position	= 22 pixels, because [(REG[17h] bits 1-0, REG[16h] bits 7-0)] = 0
HDP	= Horizontal Display Period	= [((REG[14h] bits 6-0) + 1) x 8] pixels

*For passive panels, the HDP must be a minimum of 32 pixels and must be increased by multiples of 16.
*HPS must comply with the following formula:

HPS > HDP + 22
HPS + HPW < HT

*Panel Type Bits (REG[10h] bits 1-0) = 00b (STN)
*FPFRAME Pulse Polarity Bit (REG[24h] bit 23) = 1 (active high)
*FPLINE Pulse Polarity Bit (REG[20h] bit 7) = 1 (active high)
*MOD^1 is the MOD signal when REG[11h] bits 5-0 = 0 (MOD toggles every FPFRAME)
*MOD^2 is the MOD signal when REG[11h] bits 5-0 = n (MOD toggles every n FPLINE)

6.4.2 Single Monochrome 4-Bit Panel Timing

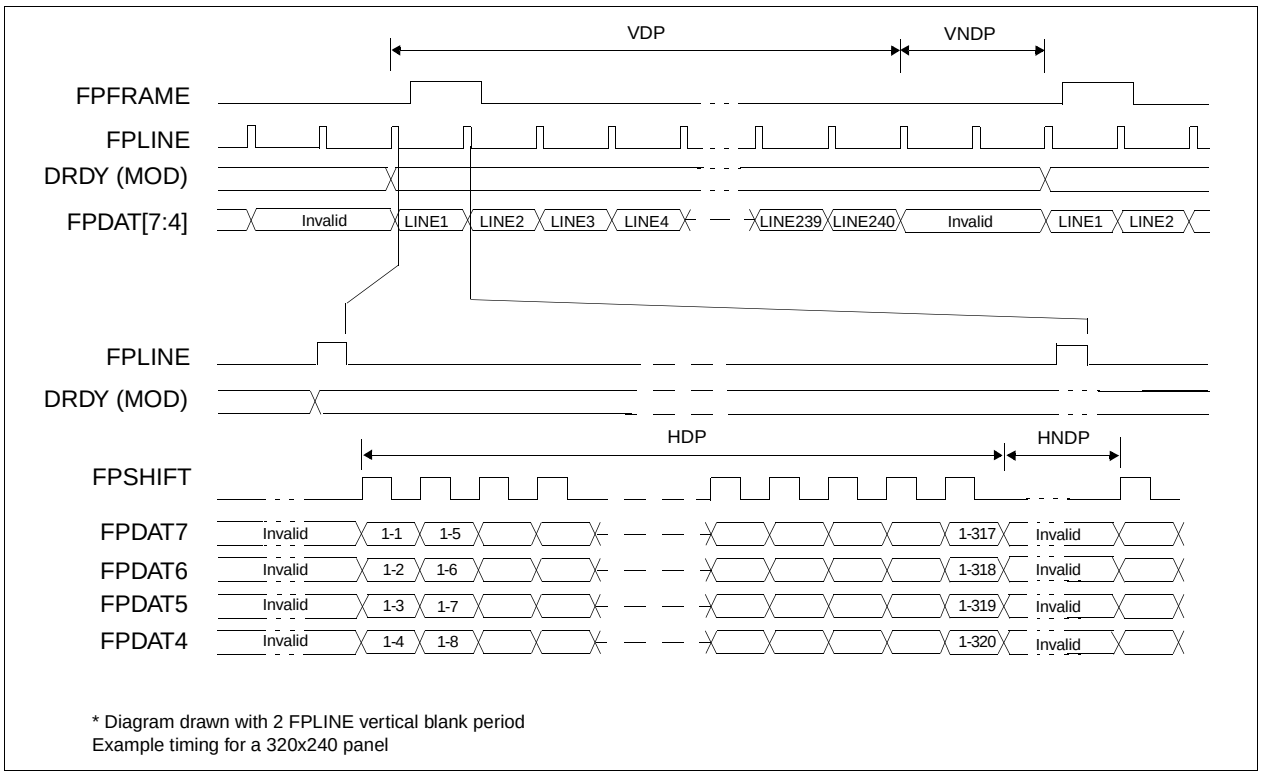


Figure 6-17 Single Monochrome 4-Bit Panel Timing

VDP = Vertical Display Period
= (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) + 1 Lines
VNDP = Vertical Non-Display Period
= VT - VDP
= (REG[19h] bits 1:0, REG[18h] bits 7:0) - (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) Lines
HDP = Horizontal Display Period
= ((REG[14h] bits 6:0) + 1) x 8Ts
HNDP = Horizontal Non-Display Period
= HT - HDP
= (((REG[12h] bits 6:0) + 1) x 8Ts) - (((REG[14h] bits 6:0) + 1) x 8Ts)

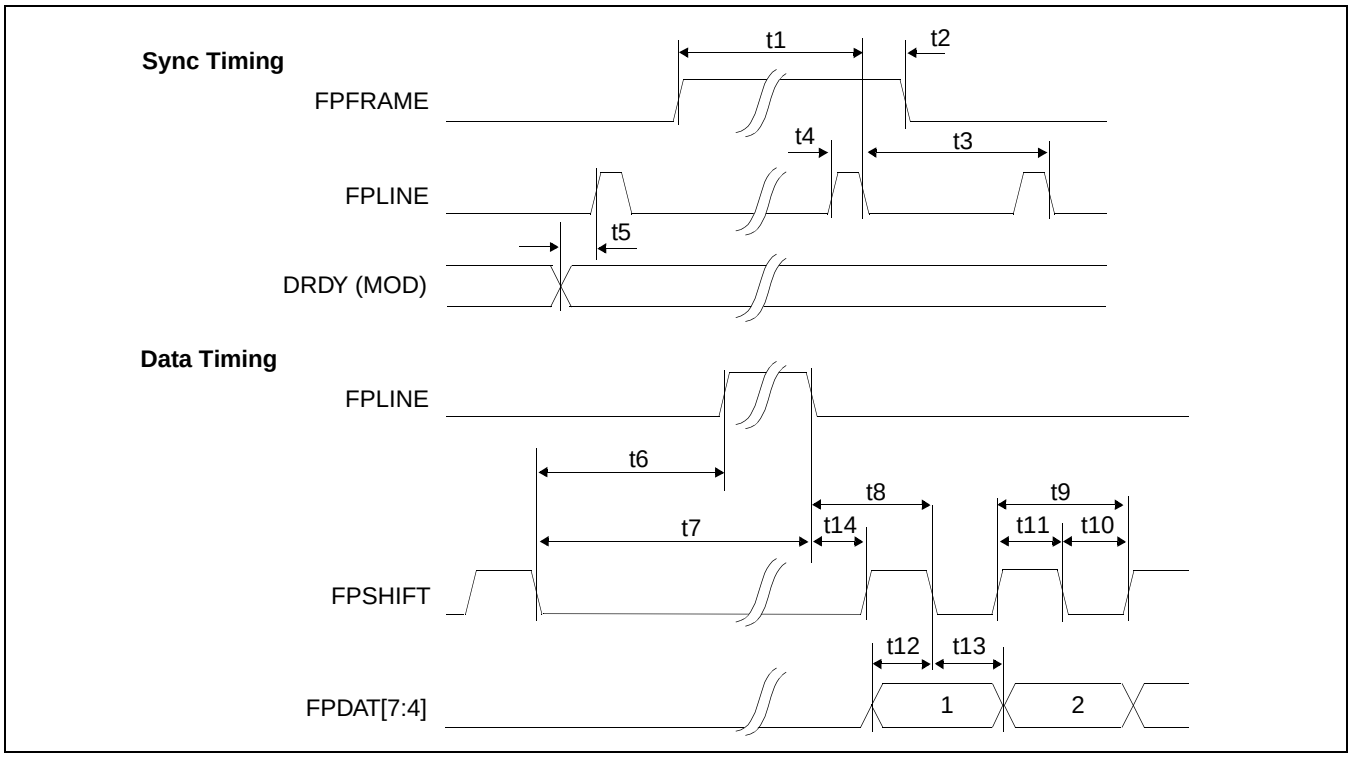


Figure 6-18 Single Monochrome 4-Bit Panel A.C. Timing

Table 6-19: Single Monochrome 4-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	note 3			Ts
t3	FPLINE period	note 4			Ts
t4	FPLINE pulse width	note 5			Ts
t5	MOD transition to FPLINE rising edge	note 6			Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 7			Ts
t7	FPSHIFT falling edge to FPLINE falling edge	t6 + t4			Ts
t8	FPLINE falling edge to FPSHIFT falling edge	t14 + 2			Ts
t9	FPSHIFT period	4			Ts
t10	FPSHIFT pulse width low	2			Ts
t11	FPSHIFT pulse width high	2			Ts
t12	FPDAT[7:4] setup to FPSHIFT falling edge	1			Ts
t13	FPDAT[7:4] hold to FPSHIFT falling edge	2			Ts
t14	FPLINE falling edge to FPSHIFT rising edge	note 8			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = HPS + $t4_{min}$
3. $t2_{min}$ = $t3_{min}$ - (HPS + $t4_{min}$)
4. $t3_{min}$ = HT
5. $t4_{min}$ = HPW
6. $t5_{min}$ = HPS - 1
7. $t6_{min}$ = HPS - (HDP + HDPS) + 2, if negative add $t3_{min}$
8. $t14_{min}$ = HDPS - (HPS + $t4_{min}$), if negative add $t3_{min}$

6.4.3 Single Monochrome 8-Bit Panel Timing

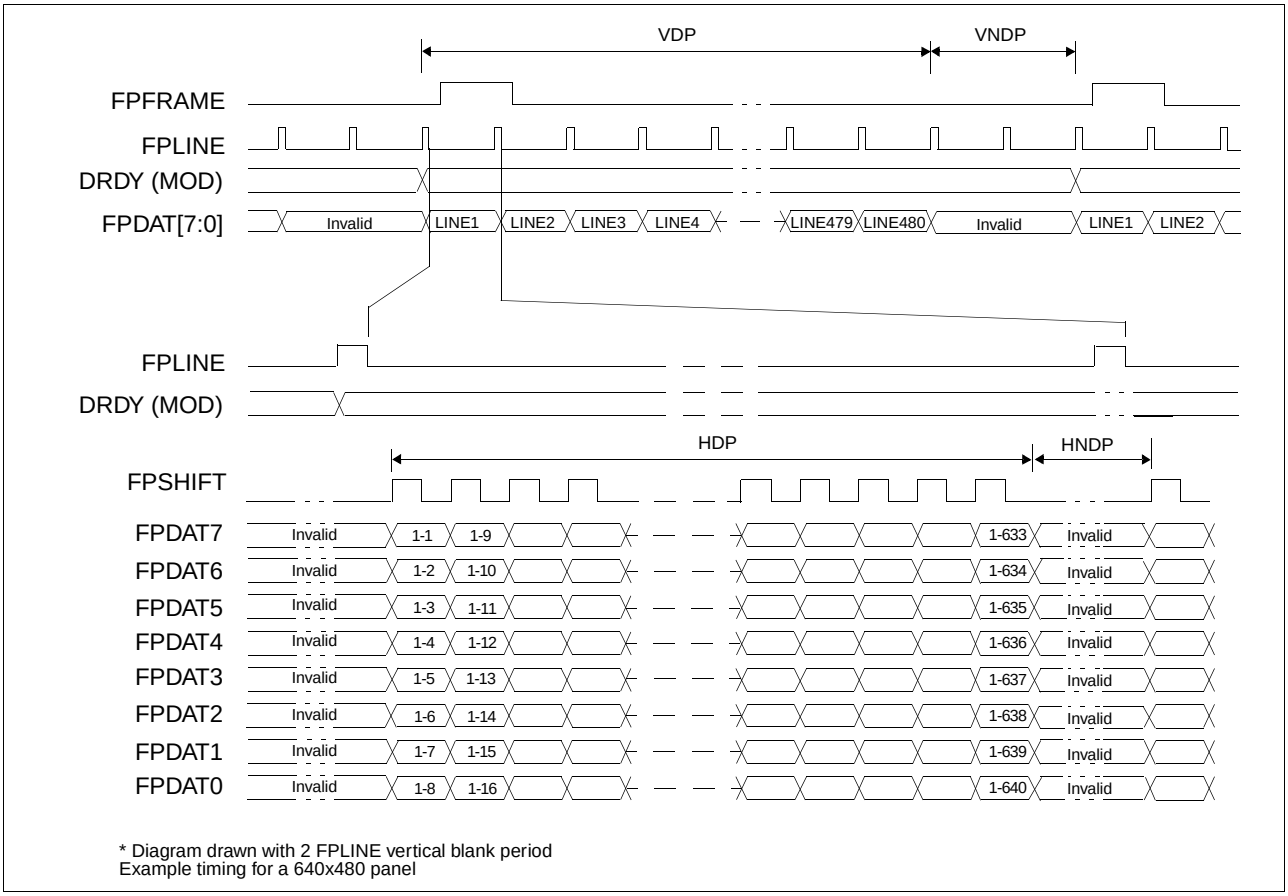


Figure 6-19 Single Monochrome 8-Bit Panel Timing

VDP = Vertical Display Period
= (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) + 1 Lines
VNDP = Vertical Non-Display Period
= VT - VDP
= (REG[19h] bits 1:0, REG[18h] bits 7:0) - (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) Lines
HDP = Horizontal Display Period
= ((REG[14h] bits 6:0) + 1) x 8Ts
HNDP = Horizontal Non-Display Period
= HT - HDP
= (((REG[12h] bits 6:0) + 1) x 8Ts) - (((REG[14h] bits 6:0) + 1) x 8Ts)

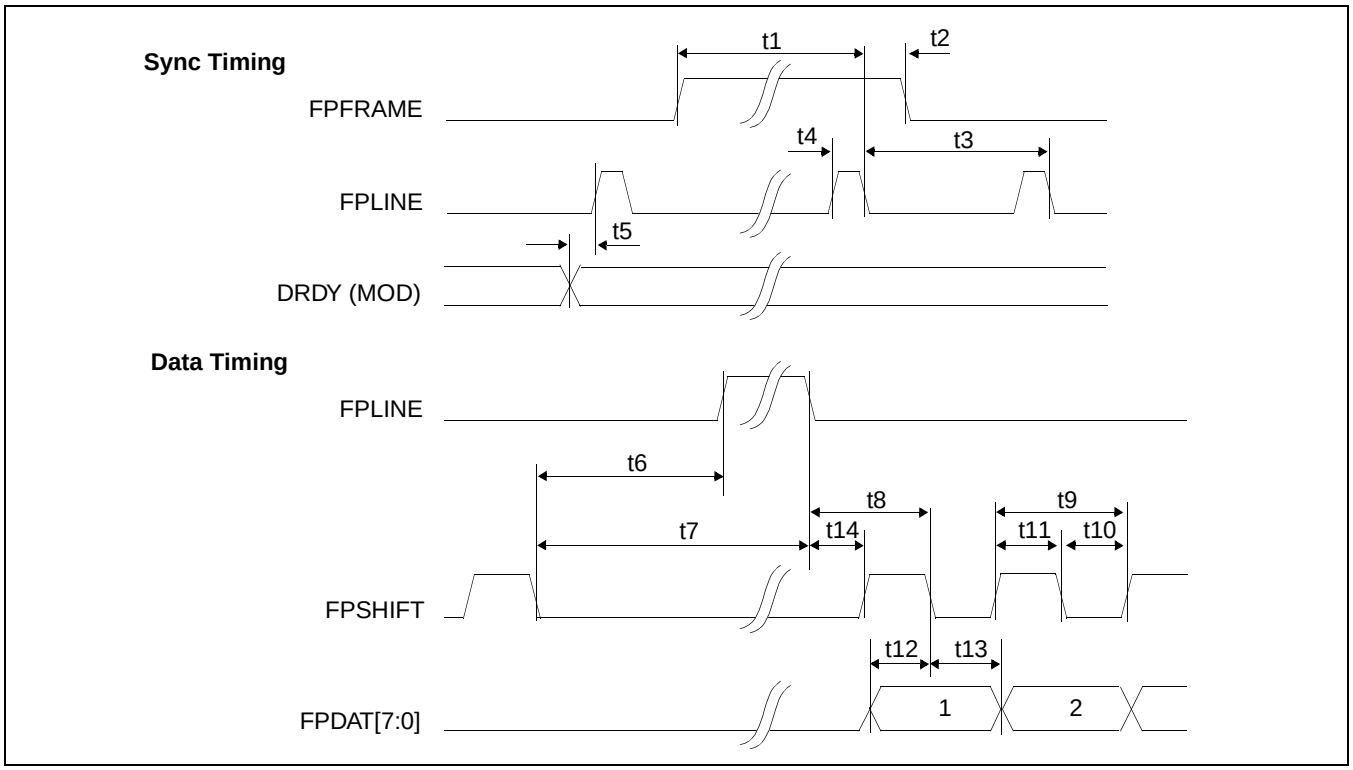


Figure 6-20 Single Monochrome 8-Bit Panel A.C. Timing

Table 6-20: Single Monochrome 8-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	note 3			Ts
t3	FPLINE period	note 4			Ts
t4	FPLINE pulse width	note 5			Ts
t5	MOD transition to FPLINE rising edge	note 6			Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 7			Ts
t7	FPSHIFT falling edge to FPLINE falling edge	t6 + t4			Ts
t8	FPLINE falling edge to FPSHIFT falling edge	t14 + 4			Ts
t9	FPSHIFT period	8			Ts
t10	FPSHIFT pulse width low	4			Ts
t11	FPSHIFT pulse width high	4			Ts
t12	FPDAT[7:0] setup to FPSHIFT falling edge	4			Ts
t13	FPDAT[7:0] hold to FPSHIFT falling edge	4			Ts
t14	FPLINE falling edge to FPSHIFT rising edge	note 8			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = HPS + $t4_{min}$
3. $t2_{min}$ = $t3_{min}$ - (HPS + $t4_{min}$)
4. $t3_{min}$ = HT
5. $t4_{min}$ = HPW
6. $t5_{min}$ = HPS - 1
7. $t6_{min}$ = HPS - (HDP + HDPS) + 4, if negative add $t3_{min}$
8. $t14_{min}$ = HDPS - (HPS + $t4_{min}$), if negative add $t3_{min}$

6.4.4 Single Color 4-Bit Panel Timing

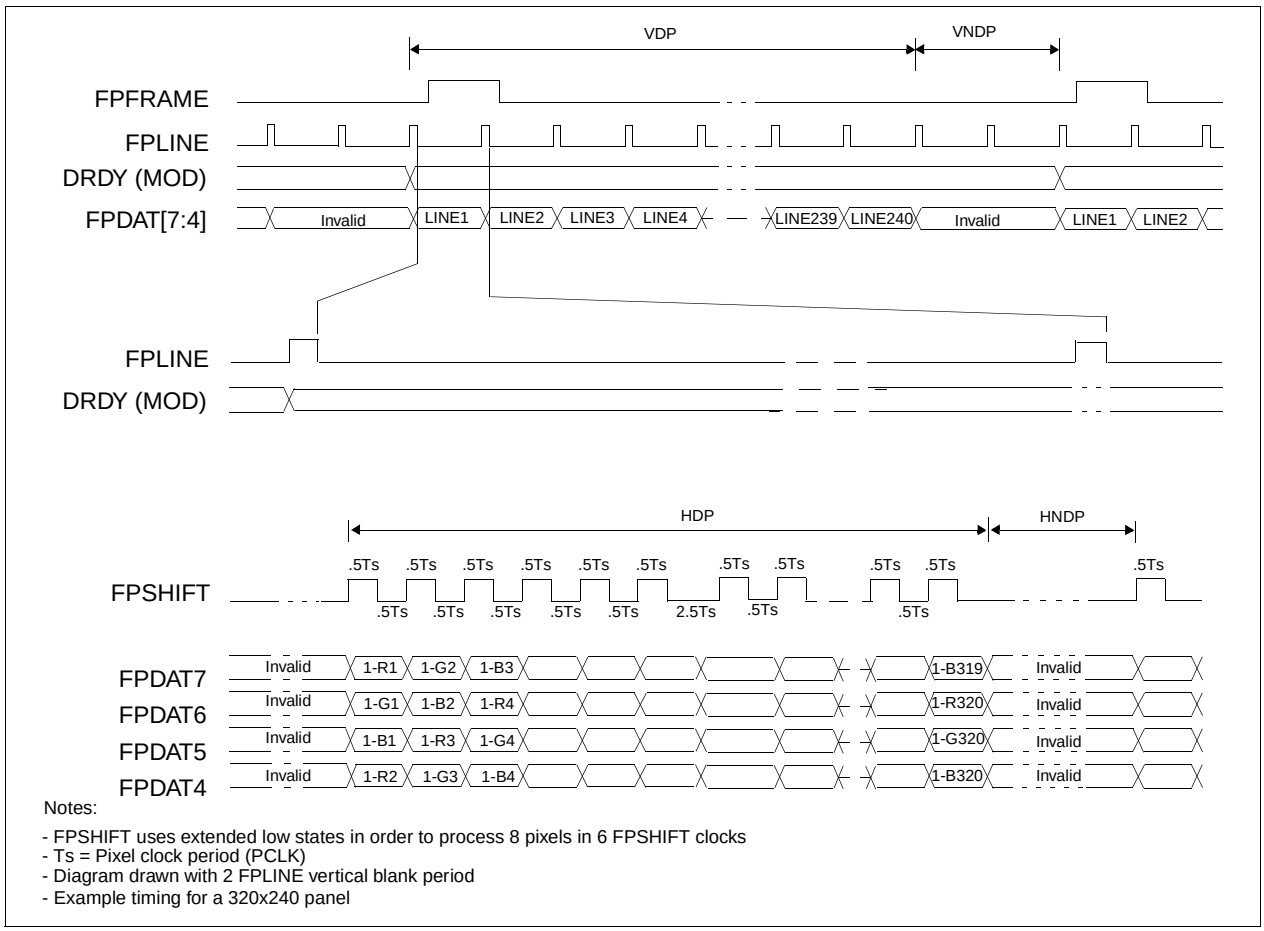


Figure 6-21 Single Color 4-Bit Panel Timing

VDP = Vertical Display Period
= (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) + 1 Lines
VNDP = Vertical Non-Display Period
= VT - VDP
= (REG[19h] bits 1:0, REG[18h] bits 7:0) - (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) Lines
HDP = Horizontal Display Period
= ((REG[14h] bits 6:0) + 1) x 8Ts
HNDP = Horizontal Non-Display Period
= HT - HDP
= (((REG[12h] bits 6:0) + 1) x 8Ts) - (((REG[14h] bits 6:0) + 1) x 8Ts)

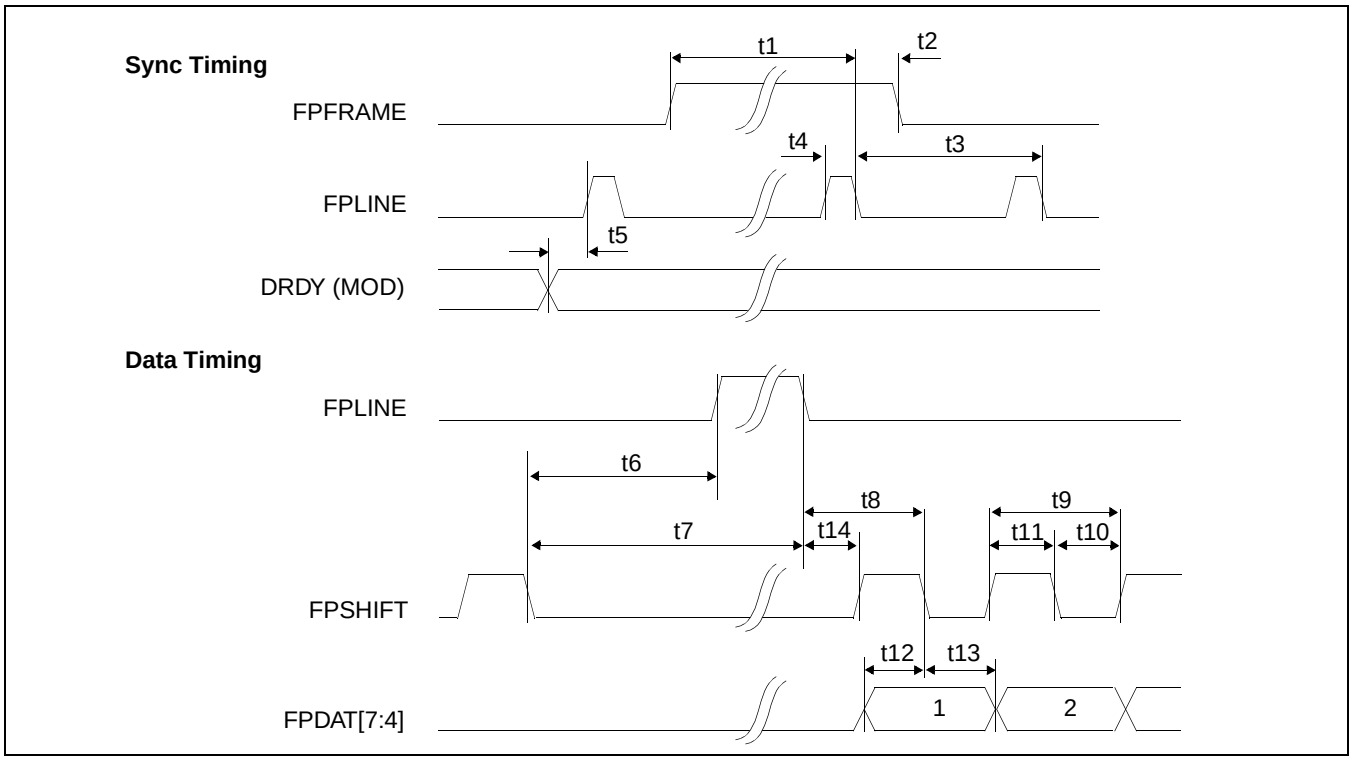


Figure 6-22 Single Color 4-Bit Panel A.C. Timing

Table 6-21: Single Color 4-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	note 3			Ts
t3	FPLINE period	note 4			Ts
t4	FPLINE pulse width	note 5			Ts
t5	MOD transition to FPLINE rising edge	note 6			Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 7			Ts
t7	FPSHIFT falling edge to FPLINE falling edge	t6 + t4			Ts
t8	FPLINE falling edge to FPSHIFT falling edge	t14 + 0.5			Ts
t9	FPSHIFT period	1			Ts
t10	FPSHIFT pulse width low	0.5			Ts
t11	FPSHIFT pulse width high	0.5			Ts
t12	FPDAT[7:4] setup to FPSHIFT falling edge	0.5			Ts
t13	FPDAT[7:4] hold to FPSHIFT falling edge	0.5			Ts
t14	FPLINE falling edge to FPSHIFT rising edge	note 8			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = HPS + $t4_{min}$
3. $t2_{min}$ = $t3_{min}$ - (HPS + $t4_{min}$)
4. $t3_{min}$ = HT
5. $t4_{min}$ = HPW
6. $t5_{min}$ = HPS - 1
7. $t6_{min}$ = HPS - (HDP + HDPS) + 1.5), if negative add $t3_{min}$
8. $t14_{min}$ = HDPS - (HPS + $t4_{min}$) + 1, if negative add $t3_{min}$

6.4.5 Single Color 8-Bit Panel Timing (Format 1)

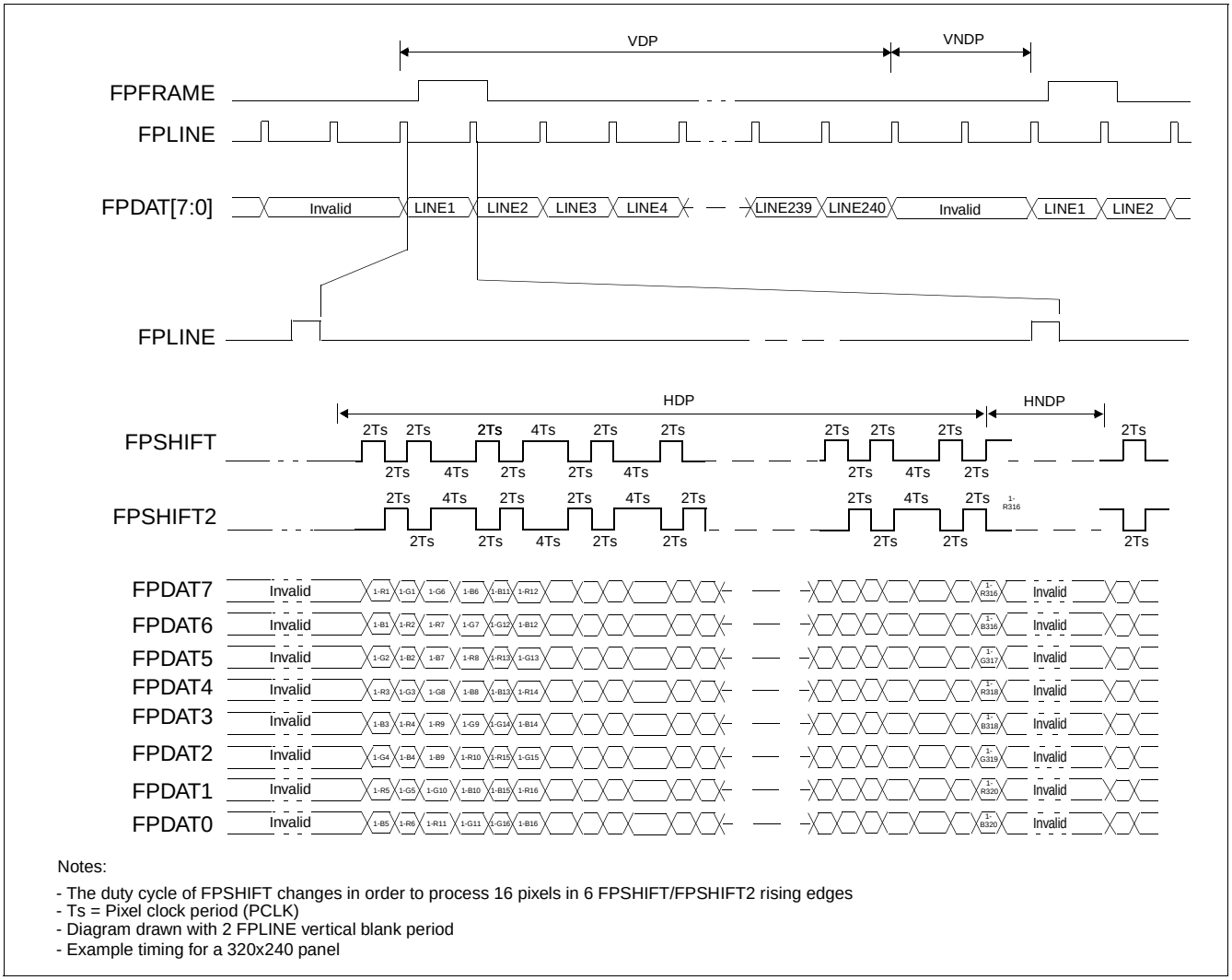


Figure 6-23 Single Color 8-Bit Panel Timing (Format 1)

VDP = Vertical Display Period
= (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) + 1 Lines
VNDP = Vertical Non-Display Period
= VT - VDP
= (REG[19h] bits 1:0, REG[18h] bits 7:0) - (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) Lines
HDP = Horizontal Display Period
= ((REG[14h] bits 6:0) + 1) x 8Ts
HNDP = Horizontal Non-Display Period
= HT - HDP
= (((REG[12h] bits 6:0) + 1) x 8Ts) - (((REG[14h] bits 6:0) + 1) x 8Ts)

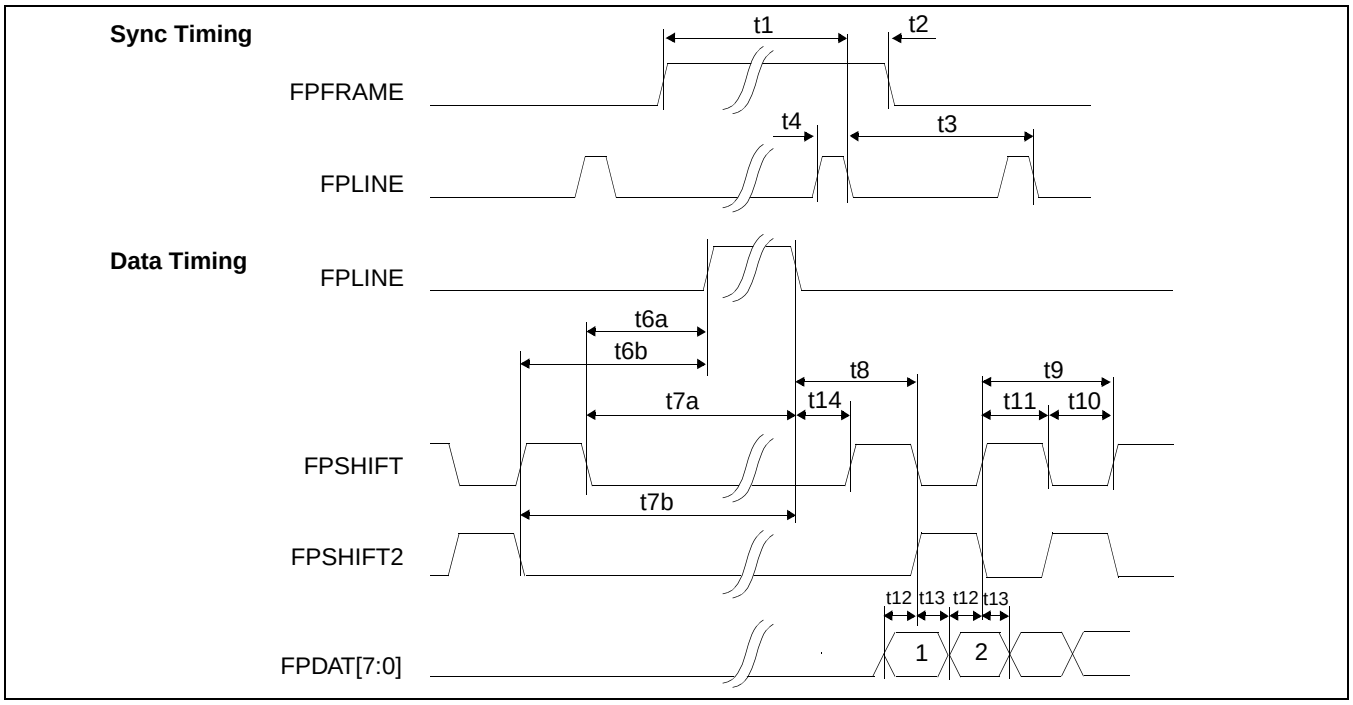


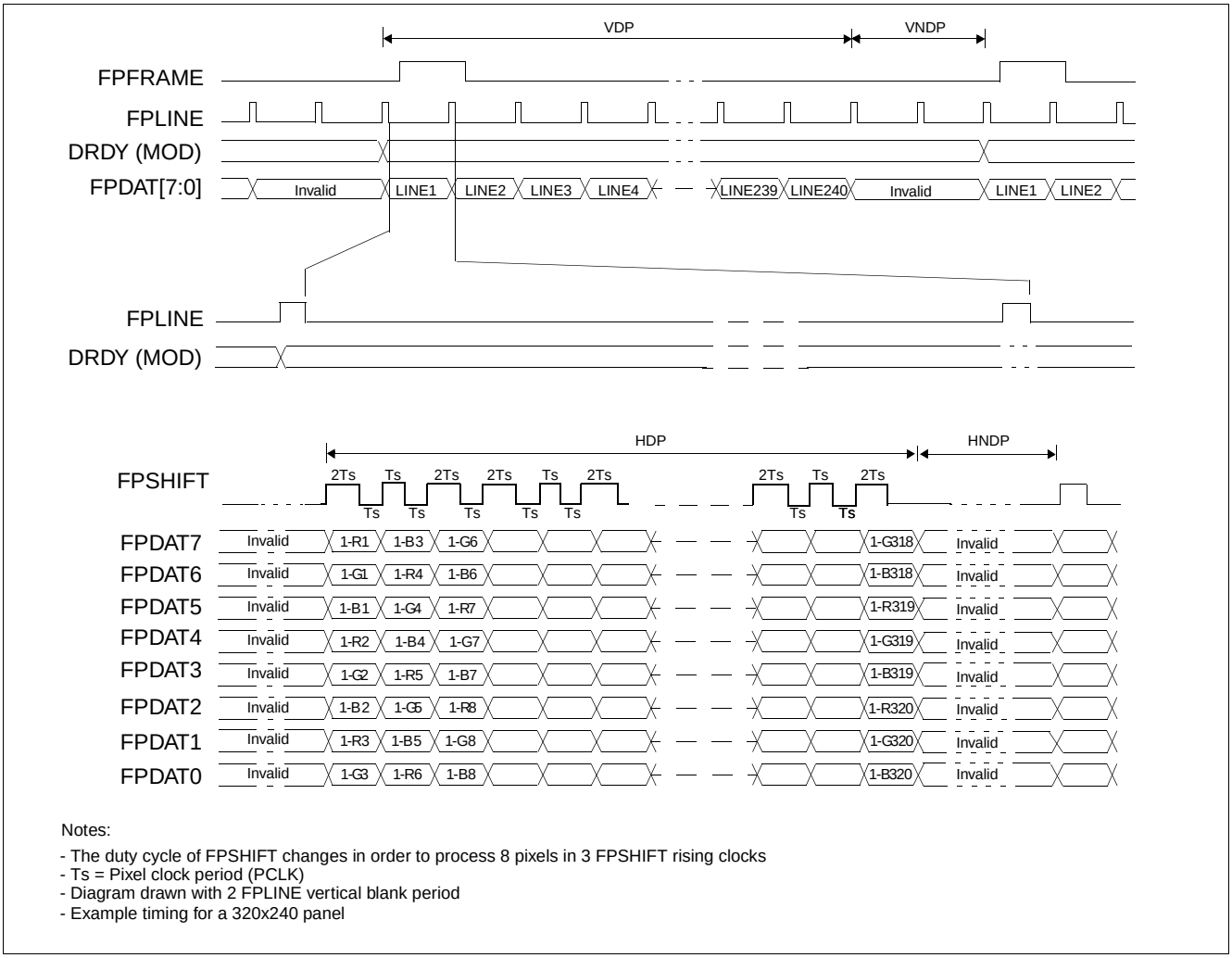
Figure 6-24 Single Color 8-Bit Panel A.C. Timing (Format 1)

Table 6-22: Single Color 8-Bit Panel A.C. Timing (Format 1)

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	note 3			Ts
t3	FPLINE period	note 4			Ts
t4	FPLINE pulse width	note 5			Ts
t6a	FPSHIFT falling edge to FPLINE rising edge	note 6			Ts
t6b	FPSHIFT2 falling edge to FPLINE rising edge	note 7			Ts
t7a	FPSHIFT falling edge to FPLINE falling edge	t6a + t4			Ts
t7b	FPSHIFT2 falling edge to FPLINE falling edge	t6b + t4			Ts
t8	FPLINE falling edge to FPSHIFT rising, FPSHIFT2 falling edge	t14 + 2			Ts
t9	FPSHIFT2, FPSHIFT period	4		6	Ts
t10	FPSHIFT2, FPSHIFT pulse width low	2			Ts
t11	FPSHIFT2, FPSHIFT pulse width high	2			Ts
t12	FPDAT[7:0] setup to FPSHIFT2, FPSHIFT falling edge	1			Ts
t13	FPDAT[7:0] hold from FPSHIFT2, FPSHIFT falling edge	1			Ts
t14	FPLINE falling edge to FPSHIFT rising edge	note 8			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = HPS + $t4_{min}$
3. $t2_{min}$ = $t3_{min}$ - (HPS + $t4_{min}$)
4. $t3_{min}$ = HT
5. $t4_{min}$ = HPW
6. $t6a_{min}$ = HPS - (HDP + HDPS), if negative add $t3_{min}$
7. $t6b_{min}$ = HPS - (HDP + HDPS) + 2, if negative add $t3_{min}$
8. $t14_{min}$ = HDPS - (HPS + $t4_{min}$), if negative add $t3_{min}$

6.4.6 Single Color 8-Bit Panel Timing (Format 2)



Notes:

- The duty cycle of FPSHIFT changes in order to process 8 pixels in 3 FPSHIFT rising clocks
- Ts = Pixel clock period (PCLK)
- Diagram drawn with 2 FPLINE vertical blank period
- Example timing for a 320x240 panel

Figure 6-25 Single Color 8-Bit Panel Timing (Format 2)

VDP = Vertical Display Period
= (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) + 1 Lines
VNDP = Vertical Non-Display Period
= VT - VDP
= (REG[19h] bits 1:0, REG[18h] bits 7:0) - (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) Lines
HDP = Horizontal Display Period
= ((REG[14h] bits 6:0) + 1) x 8Ts
HNDP = Horizontal Non-Display Period
= HT - HDP
= (((REG[12h] bits 6:0) + 1) x 8Ts) - (((REG[14h] bits 6:0) + 1) x 8Ts)

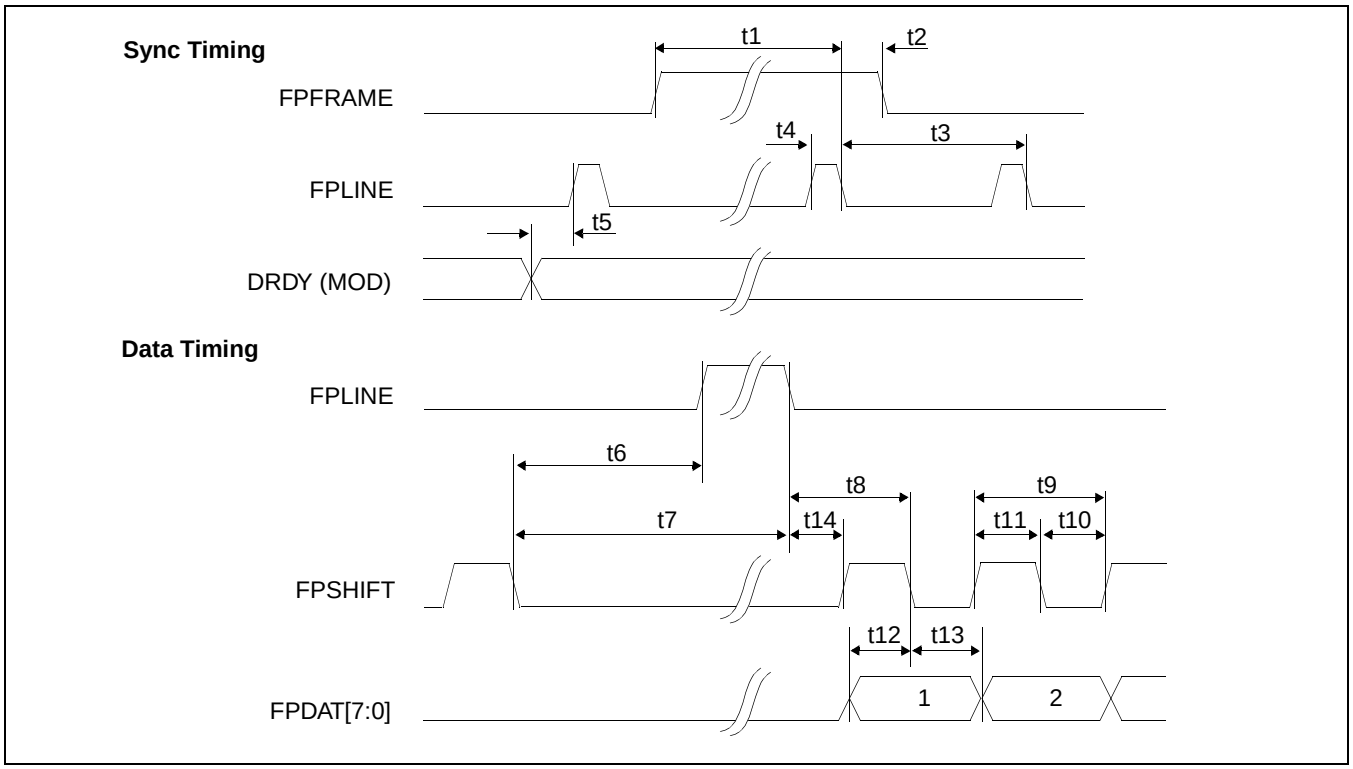


Figure 6-26 Single Color 8-Bit Panel A.C. Timing (Format 2)

Table 6-23: Single Color 8-Bit Panel A.C. Timing (Format 2)

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	note 3			Ts
t3	FPLINE period	note 4			Ts
t4	FPLINE pulse width	note 5			Ts
t5	MOD transition to FPLINE rising edge	note 6			Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 7			Ts
t7	FPSHIFT falling edge to FPLINE falling edge	t6 + t4			Ts
t8	FPLINE falling edge to FPSHIFT falling edge	t14 + 2			Ts
t9	FPSHIFT period	2			Ts
t10	FPSHIFT pulse width low	1			Ts
t11	FPSHIFT pulse width high	1			Ts
t12	FPDAT[7:0] setup to FPSHIFT falling edge	1			Ts
t13	FPDAT[7:0] hold to FPSHIFT falling edge	1			Ts
t14	FPLINE falling edge to FPSHIFT rising edge	note 8			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = HPS + $t4_{min}$
3. $t2_{min}$ = $t3_{min}$ - (HPS + $t4_{min}$)
4. $t3_{min}$ = HT
5. $t4_{min}$ = HPW
6. $t5_{min}$ = HPS - 1
7. $t6_{min}$ = HPS - (HDP + HDPS) + 1, if negative add $t3_{min}$
8. $t14_{min}$ = HDPS - (HPS + $t4_{min}$), if negative add $t3_{min}$

6.4.7 Single Color 16-Bit Panel Timing

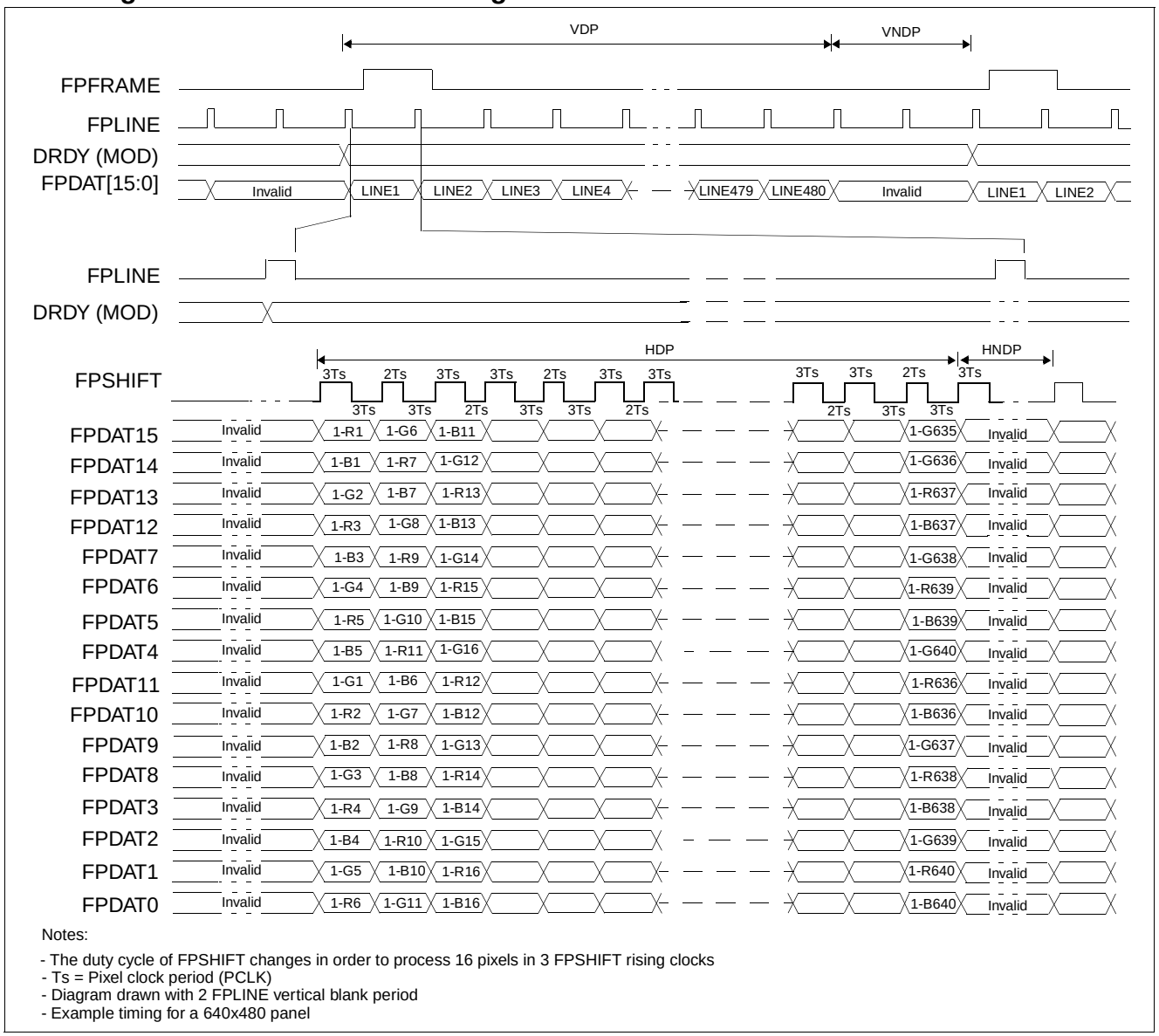


Figure 6-27 Single Color 16-Bit Panel Timing

VDP = Vertical Display Period
= (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) + 1 Lines
VNDP = Vertical Non-Display Period
= VT - VDP
= (REG[19h] bits 1:0, REG[18h] bits 7:0) - (REG[1Dh] bits 1:0, REG[1Ch] bits 7:0) Lines
HDP = Horizontal Display Period
= ((REG[14h] bits 6:0) + 1) x 8Ts
HNDP = Horizontal Non-Display Period
= HT - HDP
= (((REG[12h] bits 6:0) + 1) x 8Ts) - (((REG[14h] bits 6:0) + 1) x 8Ts)

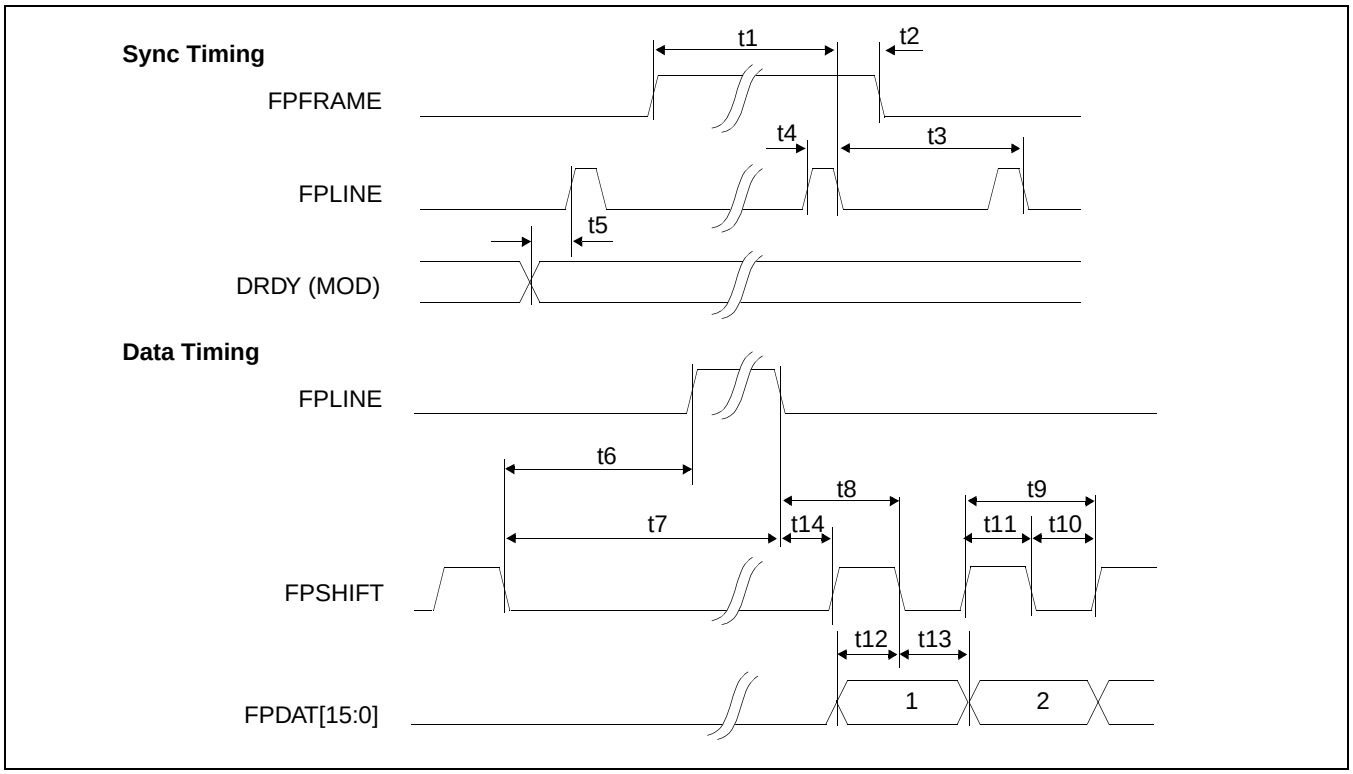


Figure 6-28 Single Color 16-Bit Panel A.C. Timing

Table 6-24: Single Color 16-Bit Panel A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME setup to FPLINE falling edge	note 2			Ts (note 1)
t2	FPFRAME hold from FPLINE falling edge	note 3			Ts
t3	FPLINE period	note 4			Ts
t4	FPLINE pulse width	note 5			Ts
t5	MOD transition to FPLINE rising edge	note 6			Ts
t6	FPSHIFT falling edge to FPLINE rising edge	note 7			Ts
t7	FPSHIFT falling edge to FPLINE falling edge	t6 + t4			Ts
t8	FPLINE falling edge to FPSHIFT falling edge	t14 + 3			Ts
t9	FPSHIFT period	5			Ts
t10	FPSHIFT pulse width low	2			Ts
t11	FPSHIFT pulse width high	2			Ts
t12	FPDAT[15:0] setup to FPSHIFT rising edge	2			Ts
t13	FPDAT[15:0] hold to FPSHIFT rising edge	2			Ts
t14	FPLINE falling edge to FPSHIFT rising edge	note 8			Ts

1. Ts = pixel clock period
2. $t1_{min}$ = HPS + $t4_{min}$
3. $t2_{min}$ = $t3_{min}$ - (HPS + $t4_{min}$)
4. $t3_{min}$ = HT
5. $t4_{min}$ = HPW
6. $t5_{min}$ = HPS - 1
7. $t6_{min}$ = HPS - (HDP + HDPS) + 2, if negative add $t3_{min}$
8. $t14_{min}$ = HDPS - (HPS + $t4_{min}$), if negative add $t3_{min}$

6.4.8 Generic TFT Panel Timing

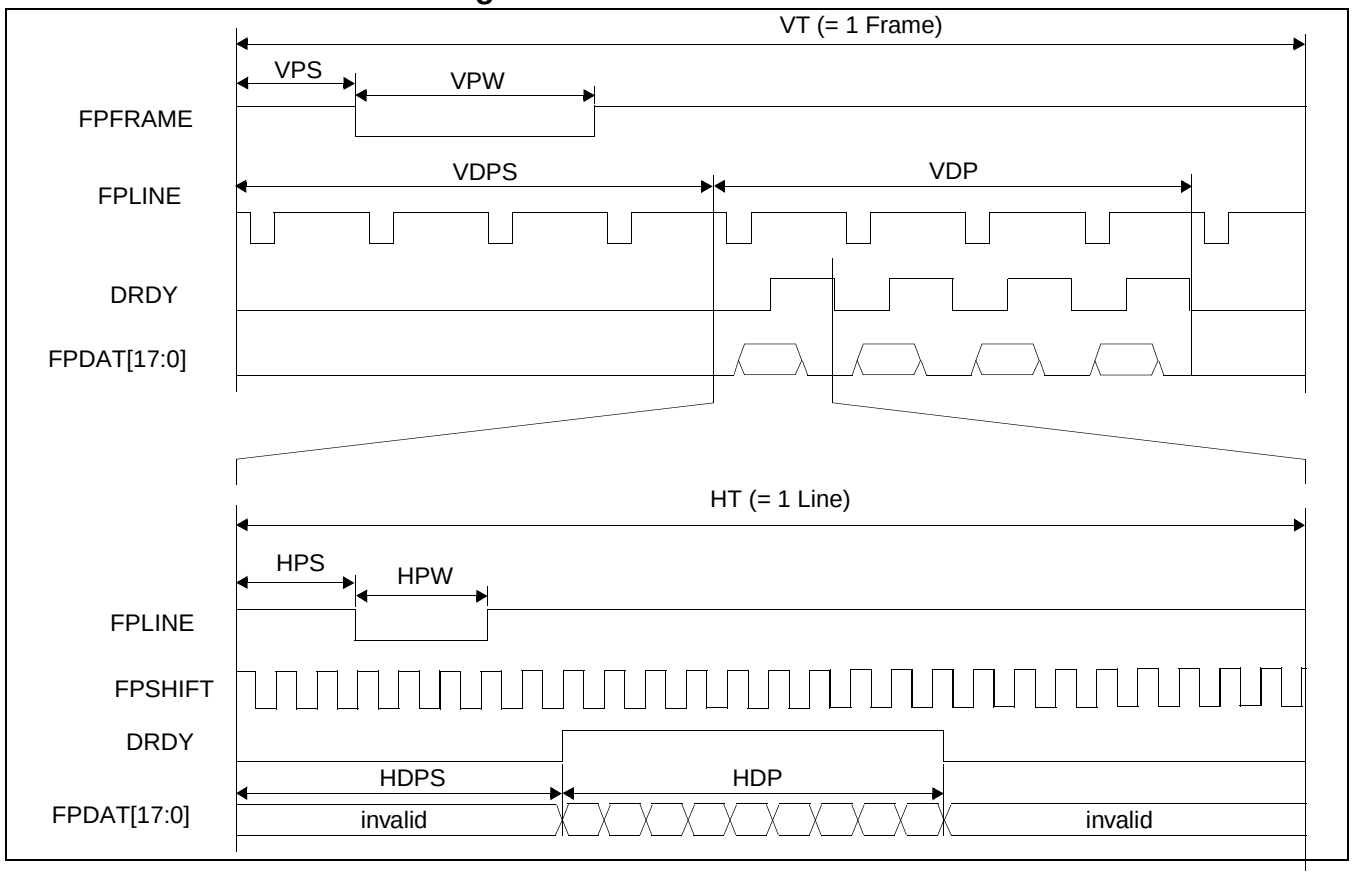


Figure 6-29 Generic TFT Panel Timing

VT	= Vertical Total	= [(REG[19h] bits 1-0, REG[18h] bits 7-0) + 1] lines
VPS	= FPFRAME Pulse Start Position	= (REG[27h] bits 1-0, REG[26h] bits 7-0) lines
VPW	= FPFRAME Pulse Width	= [(REG[24h] bits 2-0) + 1] lines
VDPS	= Vertical Display Period Start Position	= (REG[1Fh] bits 1-0, REG[1Eh] bits 7-0) lines
VDP	= Vertical Display Period	= [(REG[1Dh] bits 1-0, REG[1Ch] bits 7-0) + 1] lines
HT	= Horizontal Total	= [((REG[12h] bits 6-0) + 1) x 8] pixels
HPS	= FPLINE Pulse Start Position	= [(REG[23h] bits 1-0, REG[22h] bits 7-0) + 1] pixels
HPW	= FPLINE Pulse Width	= [(REG[20h] bits 6-0) + 1] pixels
HDPS	= Horizontal Display Period Start Position	= [(REG[17h] bits 1-0, REG[16h] bits 7-0) + 5] pixels
HDP	= Horizontal Display Period	= [((REG[14h] bits 6-0) + 1) x 8] pixels

*For TFT panels, the HDP must be a minimum of 16 pixels and must be increased by multiples of 8.
*Panel Type Bits (REG[10h] bits 1-0) = 01 (TFT)
*FPLINE Pulse Polarity Bit (REG[20h] bit 7) = 0 (active low)
*FPFRAME Polarity Bit (REG[24h] bit 7) = 0 (active low)

6.4.9 9/12/18-Bit TFT Panel Timing

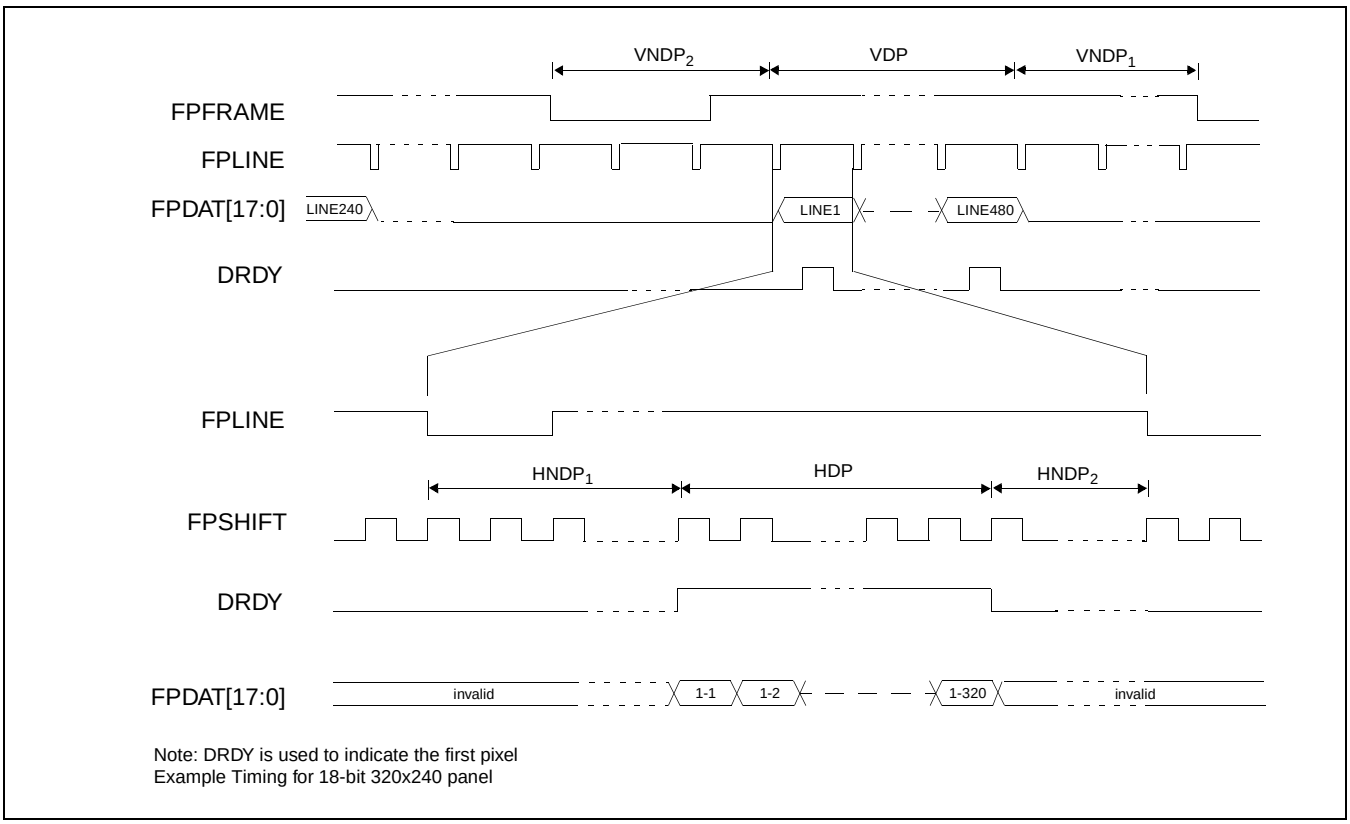


Figure 6-30 18-Bit TFT Panel Timing

VDP = Vertical Display Period
= VDP Lines
VNDP = Vertical Non-Display Period
= VNDP1 + VNDP2
= VT - VDP Lines
VNDP1 = Vertical Non-Display Period 1
= VNDP - VNDP2 Lines
VNDP2 = Vertical Non-Display Period 2
= VDPS - VPS Lines if negative add VT
HDP = Horizontal Display Period
= HDP Ts
HNDP = Horizontal Non-Display Period
= HNDP1 + HNDP2
= HT - HDP Ts
HNDP1 = Horizontal Non-Display Period 1
= HDPS - HPS Ts if negative add HT
HNDP2 = Horizontal Non-Display Period 2
= HPS - (HDP + HDPS) Ts if negative add HT

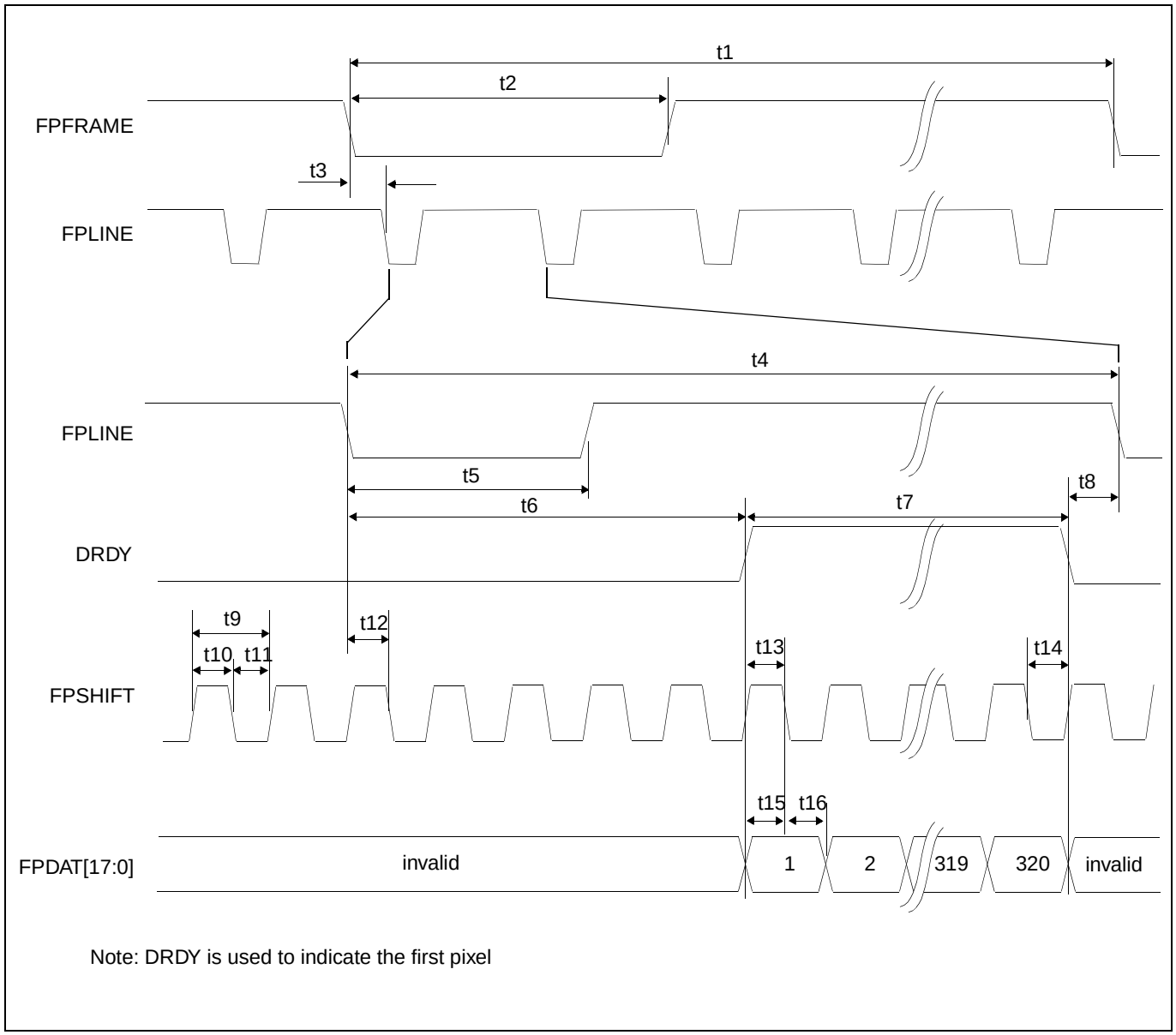


Figure 6-31 TFT A.C. Timing

Table 6-25: TFT A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME cycle time	VT			Lines
t2	FPFRAME pulse width low	VPW			Lines
t3	FPFRAME falling edge to FPLINE falling edge phase difference	HPS			Ts (note 1)
t4	FPLINE cycle time	HT			Ts
t5	FPLINE pulse width low	HPW			Ts
t6	FPLINE Falling edge to DRDY active	note 2		250	Ts
t7	DRDY pulse width	HDP			Ts
t8	DRDY falling edge to FPLINE falling edge	note 3			Ts
t9	FPSHIFT period	1			Ts
t10	FPSHIFT pulse width high	0.5			Ts
t11	FPSHIFT pulse width low	0.5			Ts
t12	FPLINE setup to FPSHIFT falling edge	0.5			Ts
t13	DRDY to FPSHIFT falling edge setup time	0.5			Ts
t14	DRDY hold from FPSHIFT falling edge	0.5			Ts
t15	Data setup to FPSHIFT falling edge	0.5			Ts
t16	Data hold from FPSHIFT falling edge	0.5			Ts

1. Ts = pixel clock period
2. t6min = HDPS - HPS if negative add HT
3. t8min = HPS - (HDP + HDPS) if negative add HT

6.4.10 160x160 Sharp 'Direct' HR-TFT Panel Timing (e.g. LQ031B1DDxx)

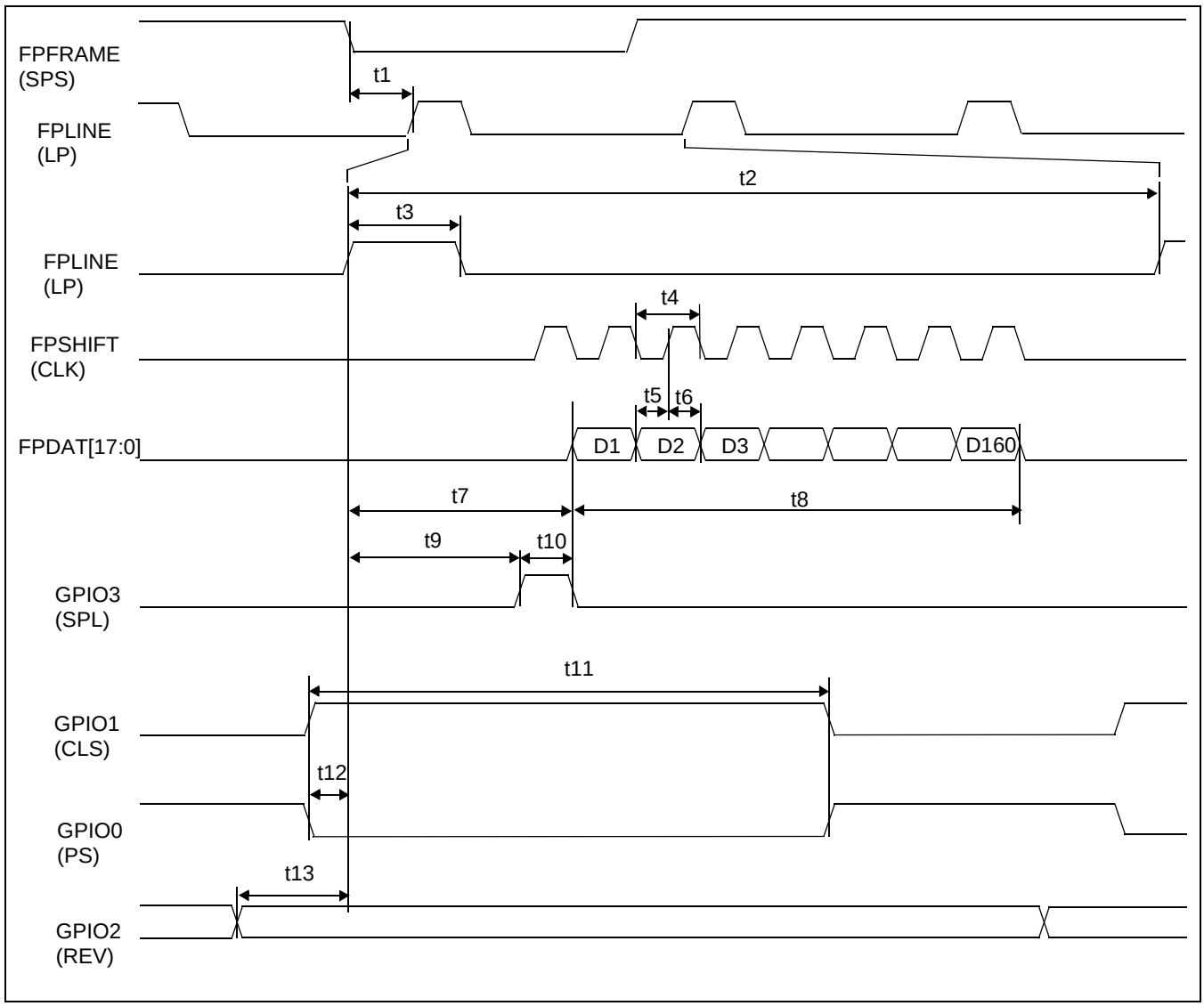


Figure 6-32 160x160 Sharp 'Direct' HR-TFT Panel Horizontal Timing

Table 6-26: 160x160 Sharp 'Direct' HR-TFT Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPLINE start position		13 (note 2)		Ts (note 1)
t2	Horizontal total period	180	note 3	220	Ts
t3	FPLINE width		2 (note 4)		Ts
t4	FPSHIFT period		1		Ts
t5	Data setup to FPSHIFT rising edge	0.5			Ts
t6	Data hold from FPSHIFT rising edge	0.5			Ts
t7	Horizontal display start position		5 (note 5)		Ts
t8	Horizontal display period		160 (note 6)		Ts
t9	FPLINE rising edge to GPIO3 rising edge		4		Ts
t10	GPIO3 pulse width		1		Ts
t11	GPIO1, GPIO0 pulse width		136		Ts
t12	GPIO1 rising edge (GPIO0 falling edge) to FPLINE rise edge		4		Ts
t13	GPIO2 toggle edge to FPLINE rise edge		10		Ts

1. Ts = pixel clock period
2. t1typ = (REG[23h] bits 1-0, REG[22h] bits 7-0) + 1
3. t2typ = ((REG[12h] bits 6-0) + 1) x 8
4. t3typ = (REG[20h] bits 6-0) + 1
5. t7typ = ((REG[17h] bits 1-0, REG[16h] bits 7-0) + 5) - ((REG[23h] bits 1-0, REG[22h] bits 7-0) + 1)
6. t8typ = (((REG[14h] bits 6-0)) + 1) x 8

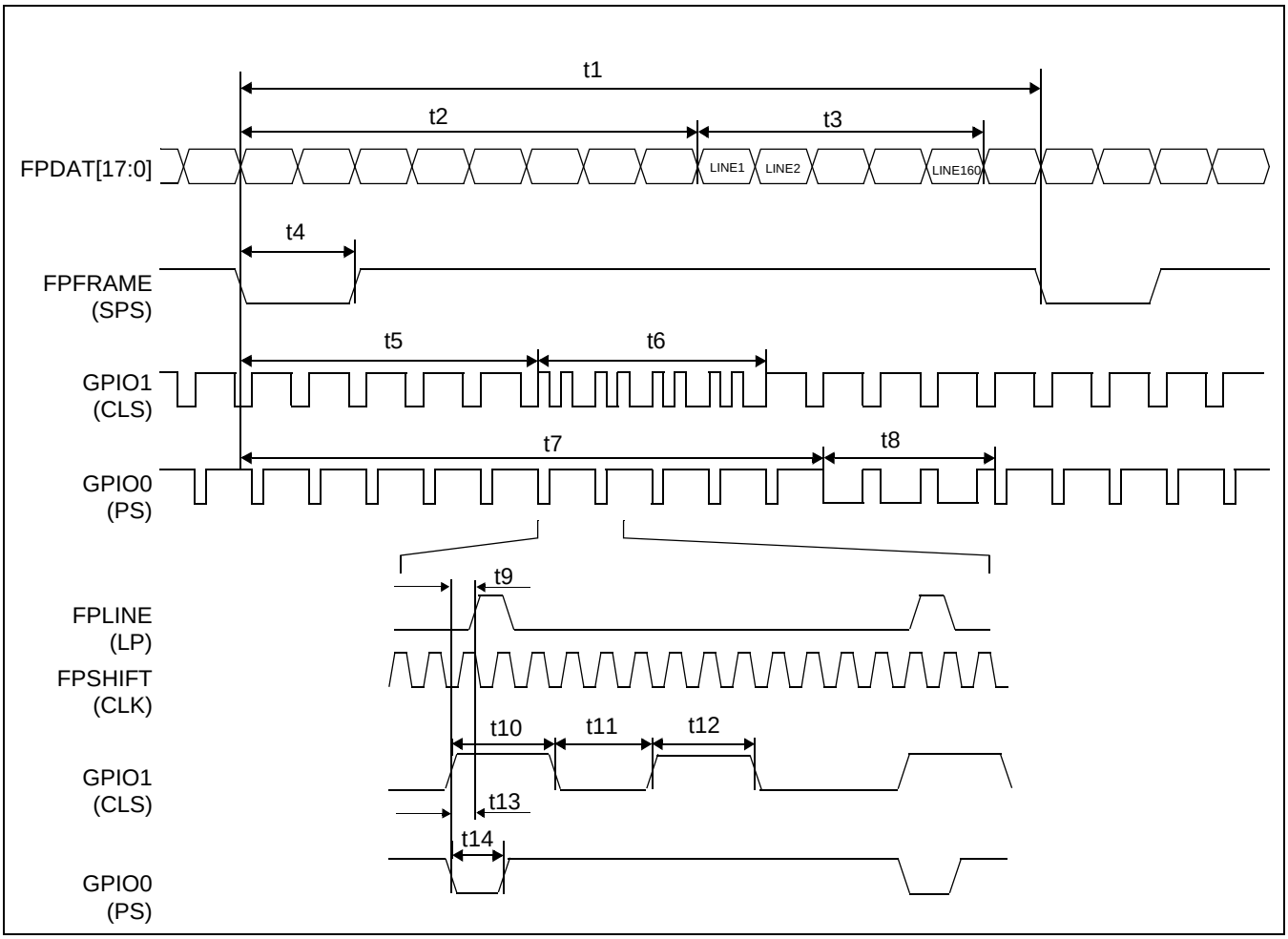


Figure 6-33 160x160 Sharp 'Direct' HR-TFT Panel Vertical Timing

Table 6-27: 160x160 Sharp 'Direct' HR-TFT Panel Vertical Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Vertical total period	203		264	Lines
t2	Vertical display start position		40		Lines
t3	Vertical display period		160		Lines
t4	Vertical sync pulse width		2		Lines
t5	FPFRAME falling edge to GPIO1 alternate timing start		5		Lines
t6	GPIO1 alternate timing period		4		Lines
t7	FPFRAME falling edge to GPIO0 alternate timing start		40		Lines
t8	GPIO0 alternate timing period		162		Lines
t9	GPIO1 first pulse rising edge to FPLINE rising edge		4		Ts (note 1)
t10	GPIO1 first pulse width		48		Ts
t11	GPIO1 first pulse falling edge to second pulse rising edge		40		Ts
t12	GPIO1 second pulse width		48		Ts
t13	GPIO0 falling edge to FPLINE rising edge		4		Ts
t14	GPIO0 low pulse width		24		Ts

1. Ts = pixel clock period

6.4.11 320x240 Sharp 'Direct' HR-TFT Panel Timing (e.g. LQ039Q2DS01)

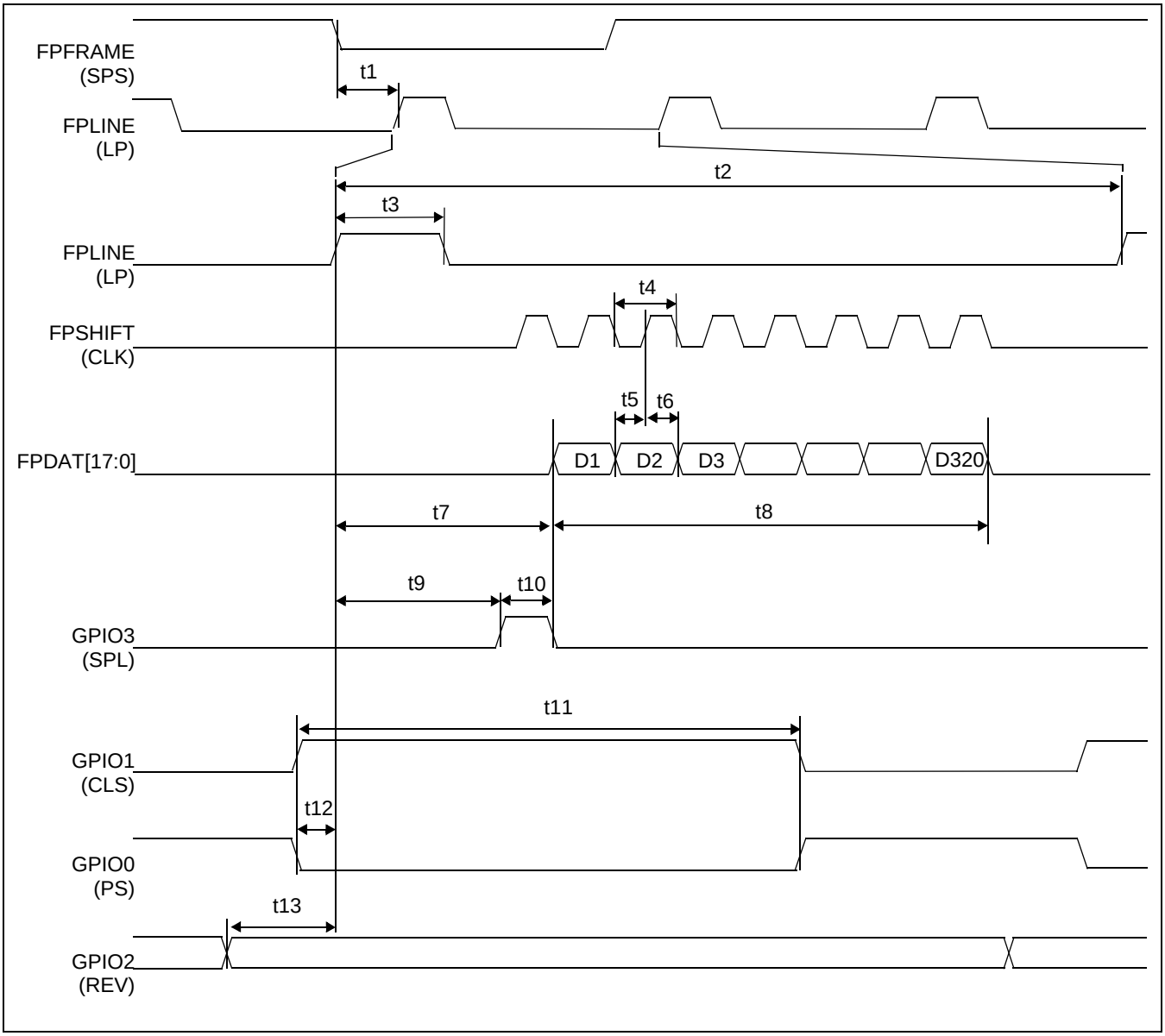


Figure 6-34 320x240 Sharp 'Direct' HR-TFT Panel Horizontal Timing

Table 6-28: 320x240 Sharp 'Direct' HR-TFT Panel Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPLINE start position		14 (note 2)		Ts (note 1)
t2	Horizontal total period	400	note 3	440	Ts
t3	FPLINE width		1 (note 4)		Ts
t4	FPSHIFT period		1		Ts
t5	Data setup to FPSHIFT rising edge	0.5			Ts
t6	Data hold from FPSHIFT rising edge	0.5			Ts
t7	Horizontal display start position		60 (note 5)		Ts
t8	Horizontal display period		320 (note 6)		Ts
t9	FPLINE rising edge to GPIO3 rising edge		59		Ts
t10	GPIO3 pulse width		1		Ts
t11	GPIO1, GPIO0 pulse width		353		Ts
t12	GPIO1 rising edge (GPIO0 falling edge) to FPLINE rise edge		5		Ts
t13	GPIO2 toggle edge to FPLINE rise edge		11		Ts

1. Ts = pixel clock period
2. t1typ = (REG[23h] bits 1-0, REG[22h] bits 7-0) + 1
3. t2typ = ((REG[12h] bits 6-0) + 1) x 8
4. t3typ = (REG[20h] bits 6-0) + 1
5. t7typ = ((REG[17h] bits 1-0, REG[16h] bits 7-0) + 5) - ((REG[23h] bits 1-0, REG[22h] bits 7-0) + 1)
6. t8typ = ((REG[14h] bits 6-0) + 1) x 8

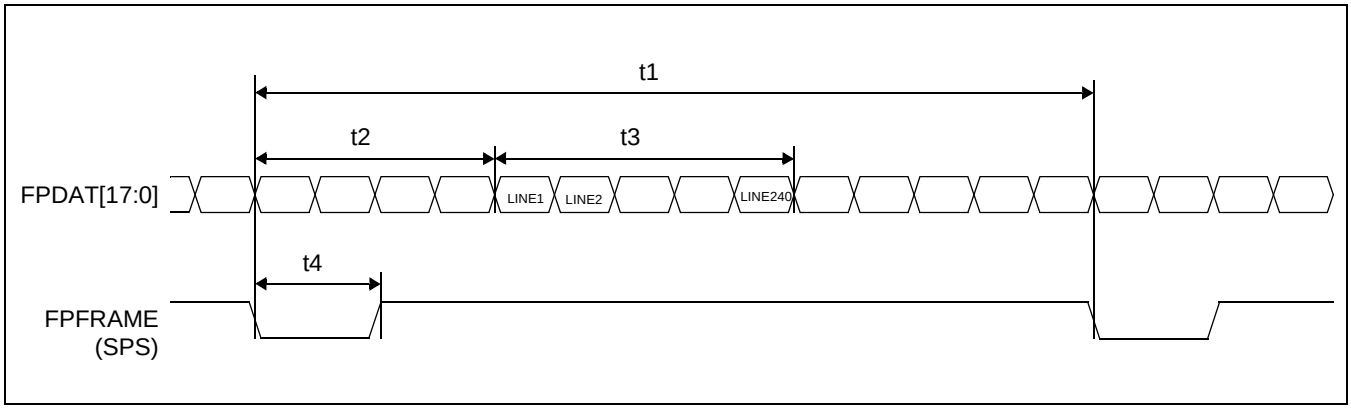


Figure 6-35 320x240 Sharp 'Direct' HR-TFT Panel Vertical Timing

Table 6-29: 320x240 Sharp 'Direct' HR-TFT Panel Vertical Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Vertical total period	245		330	Lines
t2	Vertical display start position		4		Lines
t3	Vertical display period		240		Lines
t4	Vertical sync pulse width		2		Lines

6.4.12 160x240 Epson D-TFD Panel Timing (e.g. LF26SCR)

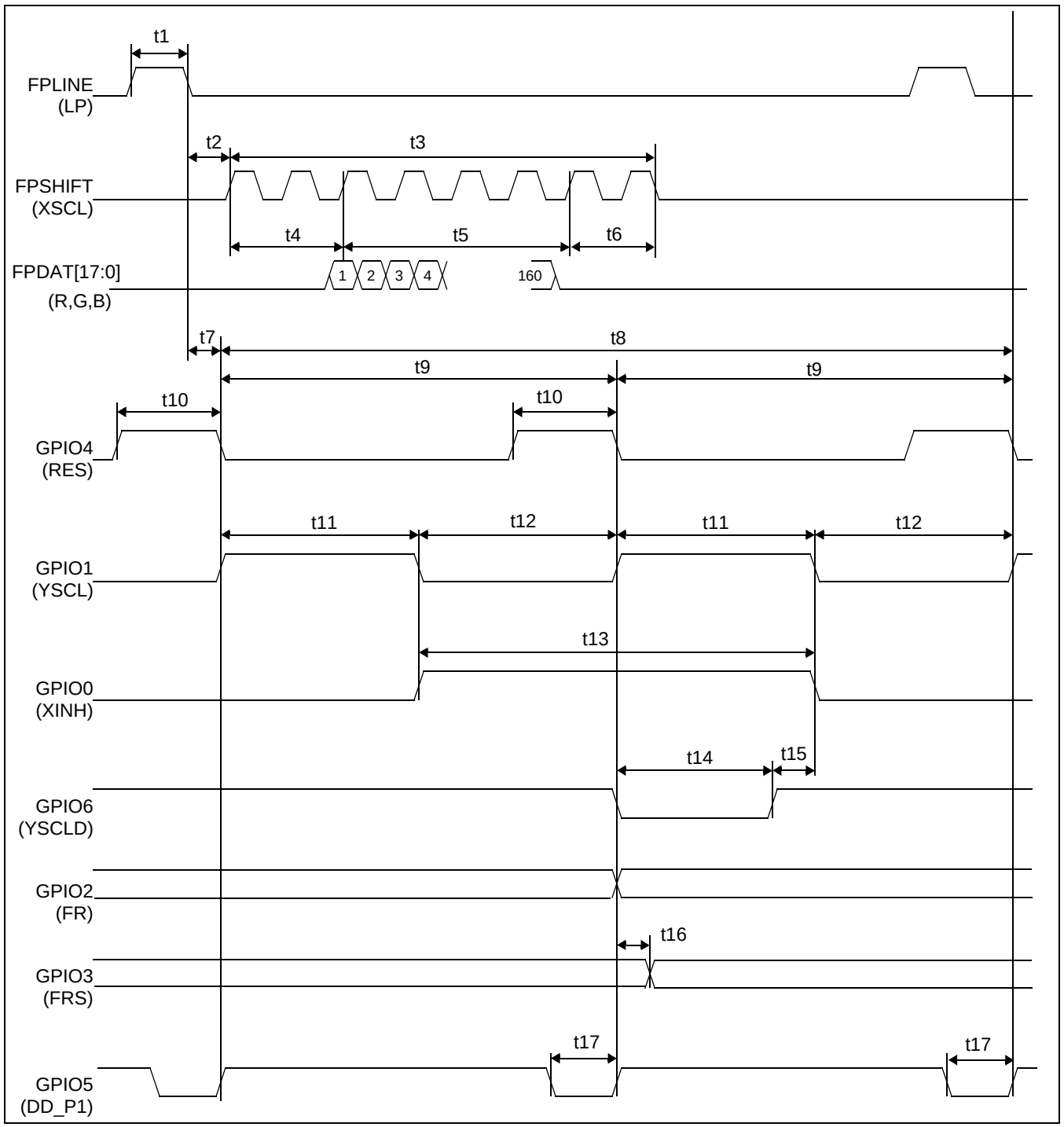


Figure 6-36 160x240 Epson D-TFD Panel Horizontal Timing

Table 6-30: 160x240 Epson D-TFD Panel Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPLINE pulse width		9		Ts (note 1)
t2	FPLINE falling edge to FPSHIFT start position		8.5		Ts
t3	FPSHIFT active period		167		Ts
t4	FPSHIFT start to first data		4		Ts
t5	Horizontal display period		160		Ts
t6	Last data to FPSHIFT inactive		3		Ts
t7	FPLINE falling edge to GPIO4 first pulse falling edge		1		Ts
t8	Horizontal total period		400		Ts
t9	GPIO4 first pulse falling edge to second pulse falling edge		200		Ts
t10	GPIO4 pulse width		11		Ts
t11	GPIO1 pulse width		100		Ts
t12	GPIO1 low period		100)		Ts
t13	GPIO0 pulse width		200		Ts
t14	GPIO6 low pulse width		90		Ts
t15	GPIO6 rising edge to GPIO0 falling edge		10		Ts
t16	GPIO2 toggle to GPIO3 toggle		1		Ts
t17	GPIO5 low pulse width		7		Ts

1. Ts = pixel clock period

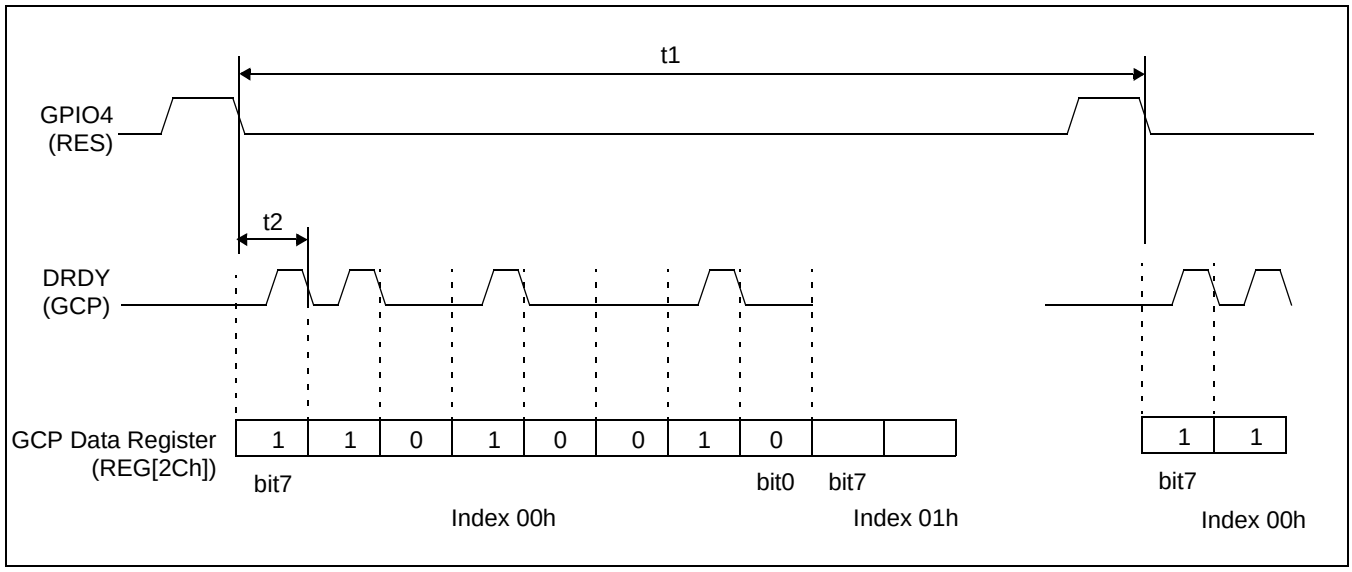


Figure 6-37 160x240 Epson D-TFD Panel GCP Horizontal Timing

Table 6-31: 160x240 Epson D-TFD Panel GCP Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Half of the horizontal total period		200		Ts (note 1)
t2	GCP clock period		1		Ts

1. Ts = pixel clock period

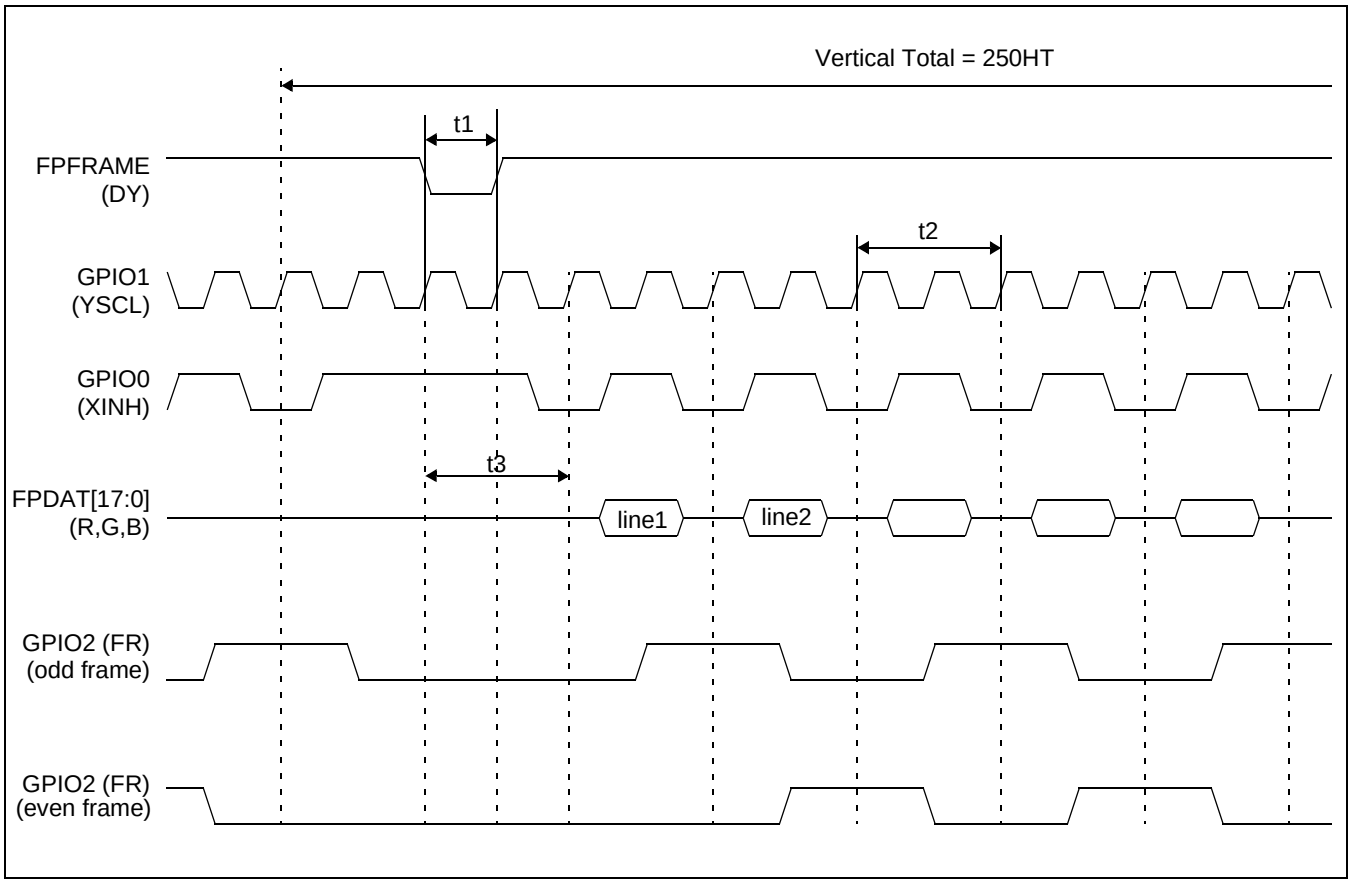


Figure 6-38 160x240 Epson D-TFD Panel Vertical Timing

Table 6-32: 160x240 Epson D-TFD Panel Vertical Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME pulse width		200		Ts (note 1)
t2	Horizontal total period		400		Ts
t3	Vertical display start		400		Ts

1. Ts = pixel clock period

6.4.13 320x240 Epson D-TFD Panel Timing (e.g. LF37SQR)

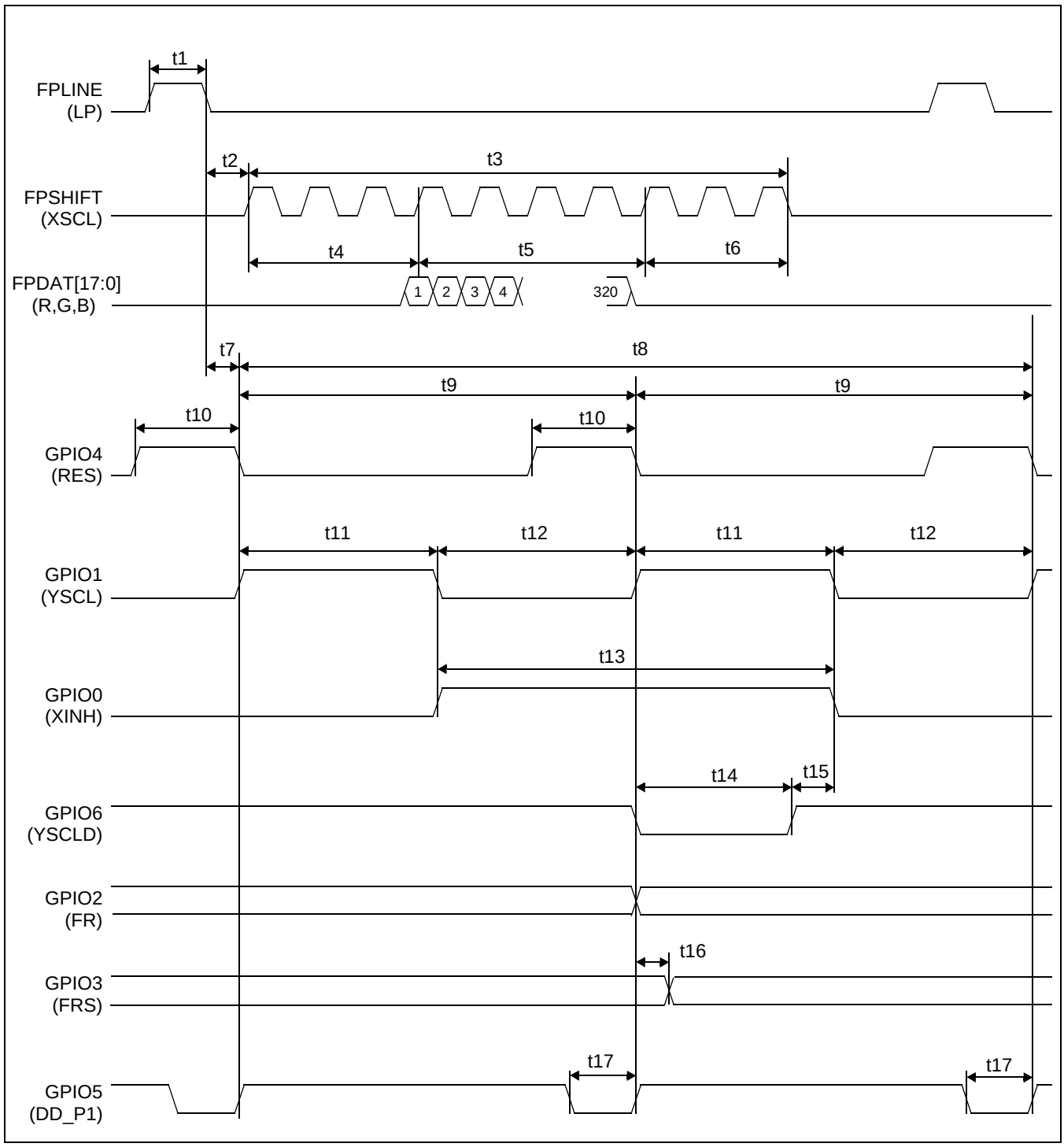


Figure 6-39 320x240 Epson D-TFD Panel Horizontal Timing

Table 6-33: 320x240 Epson D-TFD Panel Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPLINE pulse width		9		Ts (note 1)
t2	FPLINE falling edge to FPSHIFT start position		8.5		Ts
t3	FPSHIFT active period		331		Ts
t4	FPSHIFT start to first data		6		Ts
t5	Horizontal display period		320		Ts
t6	Last data to FPSHIFT inactive		5		Ts
t7	FPLINE falling edge to GPIO4 first pulse falling edge		1		Ts
t8	Horizontal total period		400		Ts
t9	GPIO4 first pulse falling edge to second pulse falling edge		200		Ts
t10	GPIO4 pulse width		11		Ts
t11	GPIO1 pulse width		100		Ts
t12	GPIO1 low period		100		Ts
t13	GPIO0 pulse width		200		Ts
t14	GPIO6 low pulse width		90		Ts
t15	GPIO6 rising edge to GPIO0 falling edge		10		Ts
t16	GPIO2 toggle to GPIO3 toggle		1		Ts
t17	GPIO5 low pulse width		7		Ts

1. Ts = pixel clock period

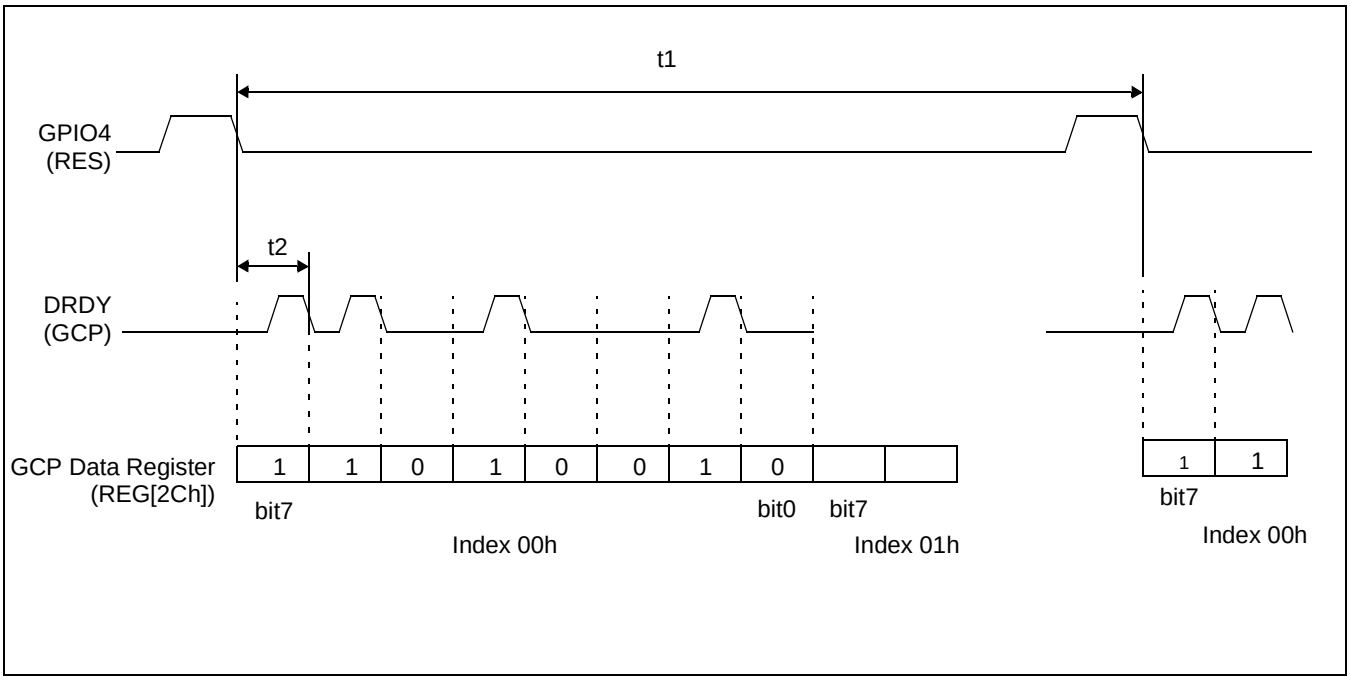


Figure 6-40 320x240 Epson D-TFD Panel GCP Horizontal Timing

Table 6-34: 320x240 Epson D-TFD Panel GCP Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Half of the horizontal total period		200		Ts (note 1)
t2	GCP clock period		1		Ts

1. Ts = pixel clock period

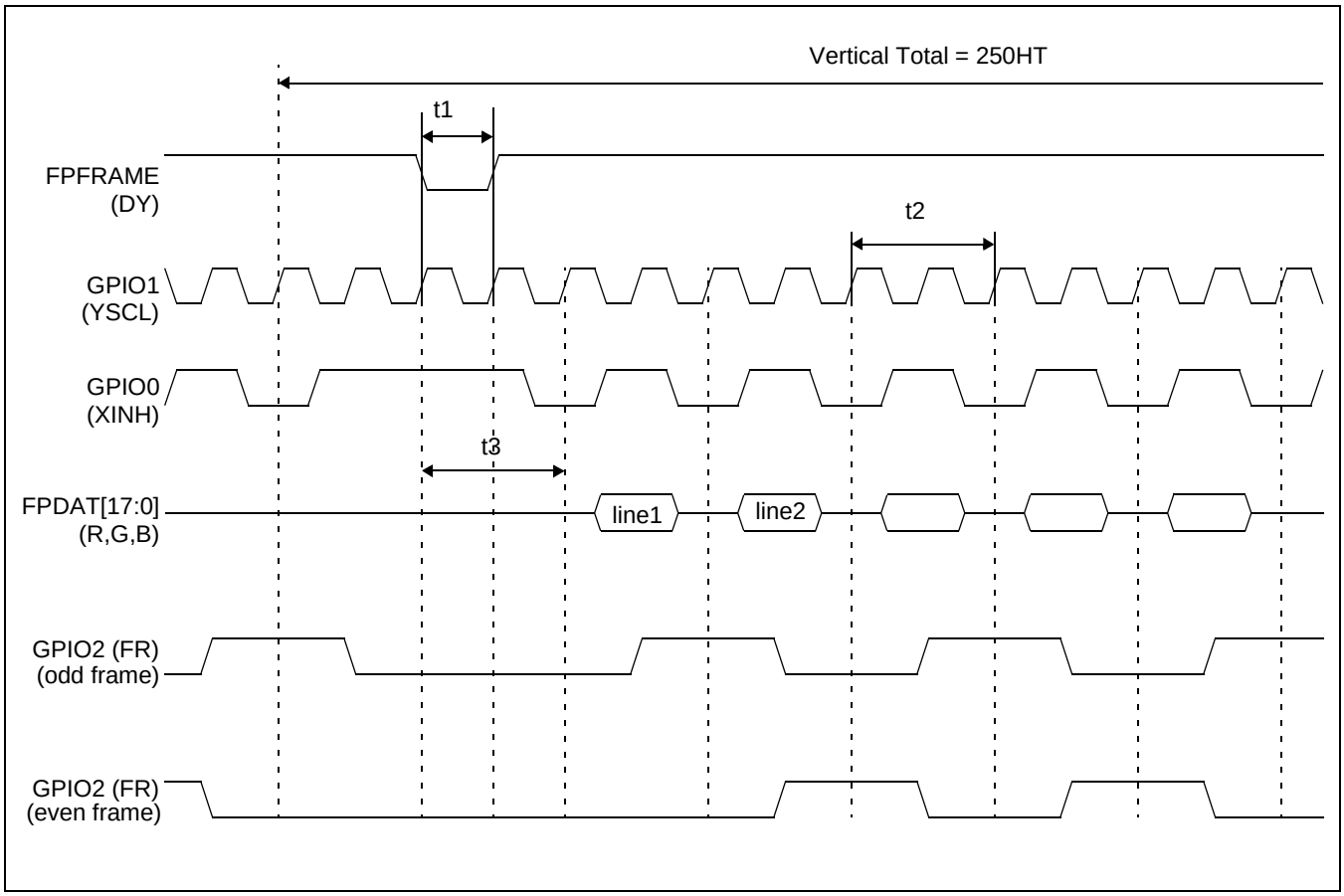


Figure 6-41 320x240 Epson D-TFD Panel Vertical Timing

Table 6-35: 320x240 Epson D-TFD Panel Vertical Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME pulse width		200		Ts (note 1)
t2	Horizontal total period		400		Ts
t3	Vertical display start		400		Ts

1. Ts = pixel clock period

6.4.14 TFT Type 2 Panel Timing

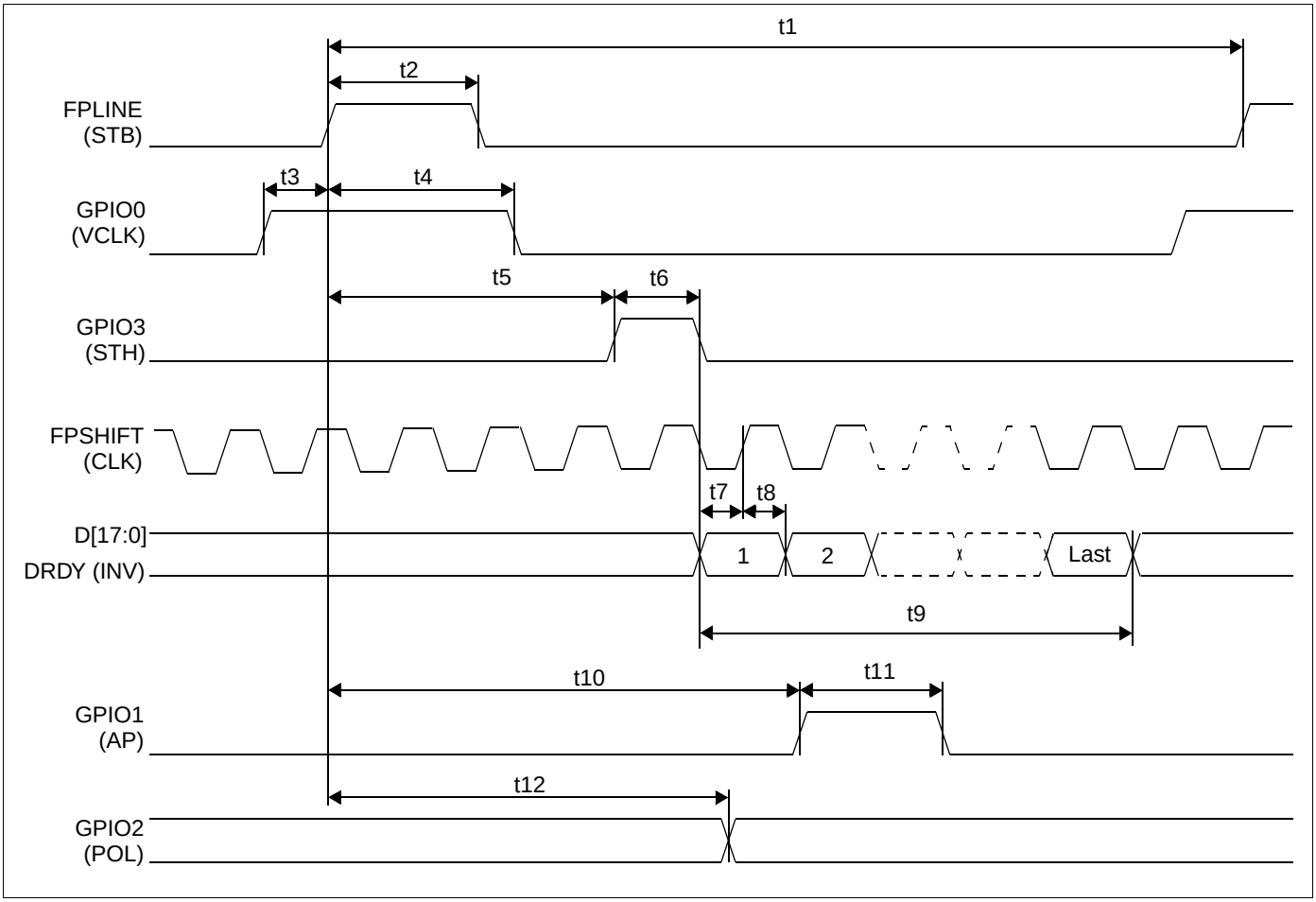


Figure 6-42 TFT Type 2 Horizontal Timing

Table 6-36: TFT Type 2 Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Horizontal total period	1	note 2	1024	Ts (note 1)
t2	FPLINE pulse width		5		Ts
t3	GPIO0 rising edge to FPLINE rising edge	7	note 3	16	Ts
t4	FPLINE rising edge to GPIO0 falling edge	7	note 4	16	Ts
t5	FPLINE rising edge to GPIO3 rising edge		note 5		Ts
t6	GPIO3 pulse width		1		Ts
t7	Data setup time	0.5			Ts
t8	Data hold time	0.5			Ts
t9	Horizontal display period	8	note 6	1024	Ts
t10	FPLINE rising edge to GPIO1 rising edge	40	note 7	90	Ts
t11	GPIO1 pulse width	20	note 8	270	Ts
t12	FPLINE rising edge to GPIO2 toggle position		10		Ts

1. Ts = pixel clock period
2. t1typ = (REG[12h] bits 6-0) +1) x 8
3. t3typ = Selected from 7, 9, 12 or 16 Ts in REG[D0h] bits 1-0
4. t4typ = Selected from 7, 9, 12 or 16 Ts in REG[D0h] bits 4-3
5. t5typ = (REG[17h] bits 1-0, REG[16h] bits 7-0) + 3
6. t9typ = ((REG[14h] bits 6-0) +1) x 8
7. t10typ = Selected from 40, 52, 68 or 90 Ts in REG[D1h] bits 1-0
8. t11typ = Selected from 20, 40, 80, 120, 150, 190, 240 or 270 Ts in REG[D1h] bits 5-3

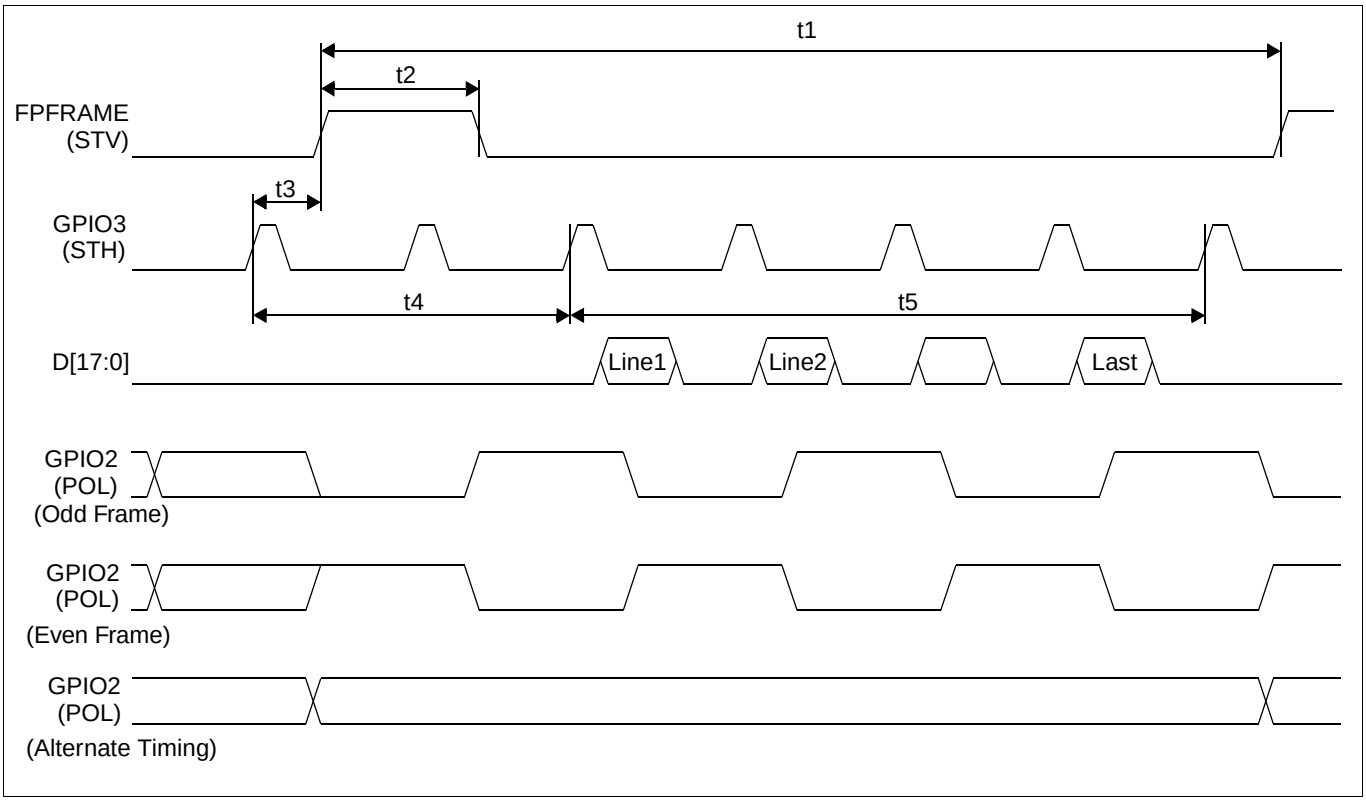


Figure 6-43 TFT Type 2 Vertical Timing

Table 6-37: TFT Type 2 Vertical Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Vertical total period	8	note 1	1024	Lines
t2	FPFRAME pulse width		1		Lines
t3	GPIO3 rising edge to FPFRAME rising edge		0		Ts (note 1)
t4	Vertical display start position	0	note 2	1024	Lines
t5	Vertical display period	1	note 3	1024	Ts

1. Ts = pixel clock period
2. t1typ = (REG[19h] bits 1-0, REG[18h] bits 7-0) +1
3. t4typ = (REG[1Fh] bits 1-0, REG[1Eh] bits 7-0)
4. t5typ = (REG[1Dh] bits 1-0], REG[1Ch] bits 7-0) + 1

6.4.15 TFT Type 3 Panel Timing

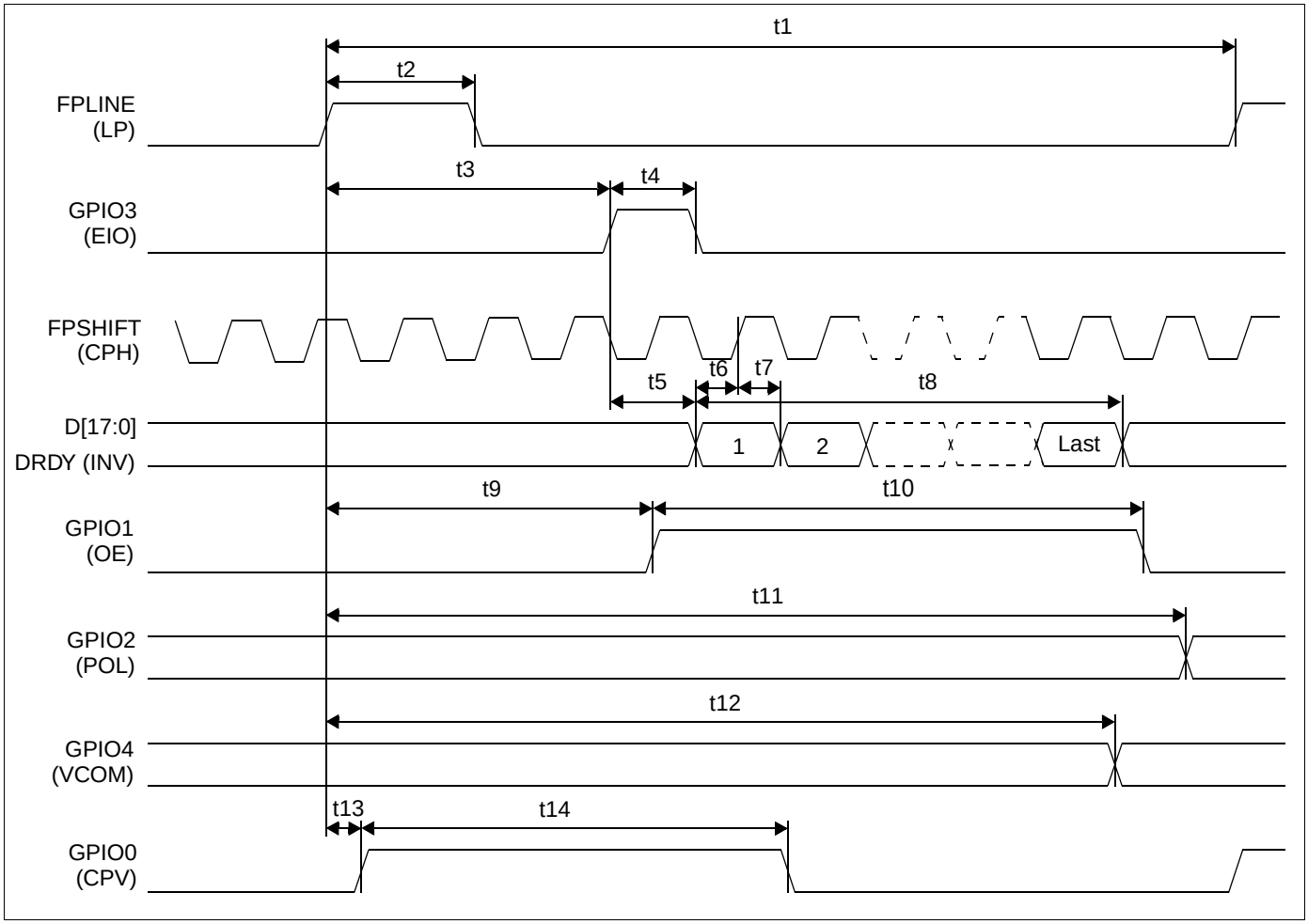


Figure 6-44 TFT Type 3 Horizontal Timing

Table 6-38: TFT Type 3 Horizontal Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Horizontal total period	8	note 2	1024	Ts (note 1)
t2	FPLINE pulse width	1	note 3	256	Ts
t3	FPLINE rising edge to GPIO3 rising edge		note 4		Ts
t4	GPIO3 pulse width				Ts
t5	GPIO3 rising edge to 1st data		1		Ts
t6	Data setup time	0.5			Ts
t7	Data hold time	0.5			Ts
t8	Horizontal display period	8	note 5	1024	Ts
t9	FPLINE rising edge to GPIO1 rising edge	0	note 6	512	Ts
t10	GPIO1 pulse width	2	note 7	512	Ts
t11	FPLINE rising edge to GPIO2 toggle position	0	note 8	512	Ts
t12	FPLINE rising edge to GPIO4 toggle position	0	note 9	512	Ts
t13	FPLINE rising edge to GPIO0 rising edge		0		Ts
t14	GPIO0 pulse width	2	note 10	512	Ts

1. Ts = pixel clock period
2. t1typ = ((REG[12h] bits 6-0) + 1) x 8
3. t2typ = (REG[20h] bits 6-0) + 1
4. t3typ = (REG[17h] bits 1-0, REG[16h] bits 7-0) + 3
5. t8typ = (REG[14h] bits 6-0) x 8
6. t9typ = ((REG[D5h] bits 7-0) x 2
7. t10typ = ((REG[D6h] bits 7-0) x 2
8. t11typ = ((REG[D7h] bits 7-0) x 2
9. t12typ = ((REG[D8h] bits 7-0) x 2
10. t14typ = ((REG[D9h] bits 7-0) x 2

Note

When TFT type 3 panel interface is selected, the following formula must be satisfied.

HTP - (HDS + HDP) > (Number of Source Driver IC * 3) + 3

Note

When REG[D6h] = 0, GPIO1 (OE) pulse width equals FPLINE (LP) pulse width.

Note

When REG[D9h] = 0, t13 = 4ns and GPIO0 (CPV) pulse width equals FPLINE (LP) pulse width.

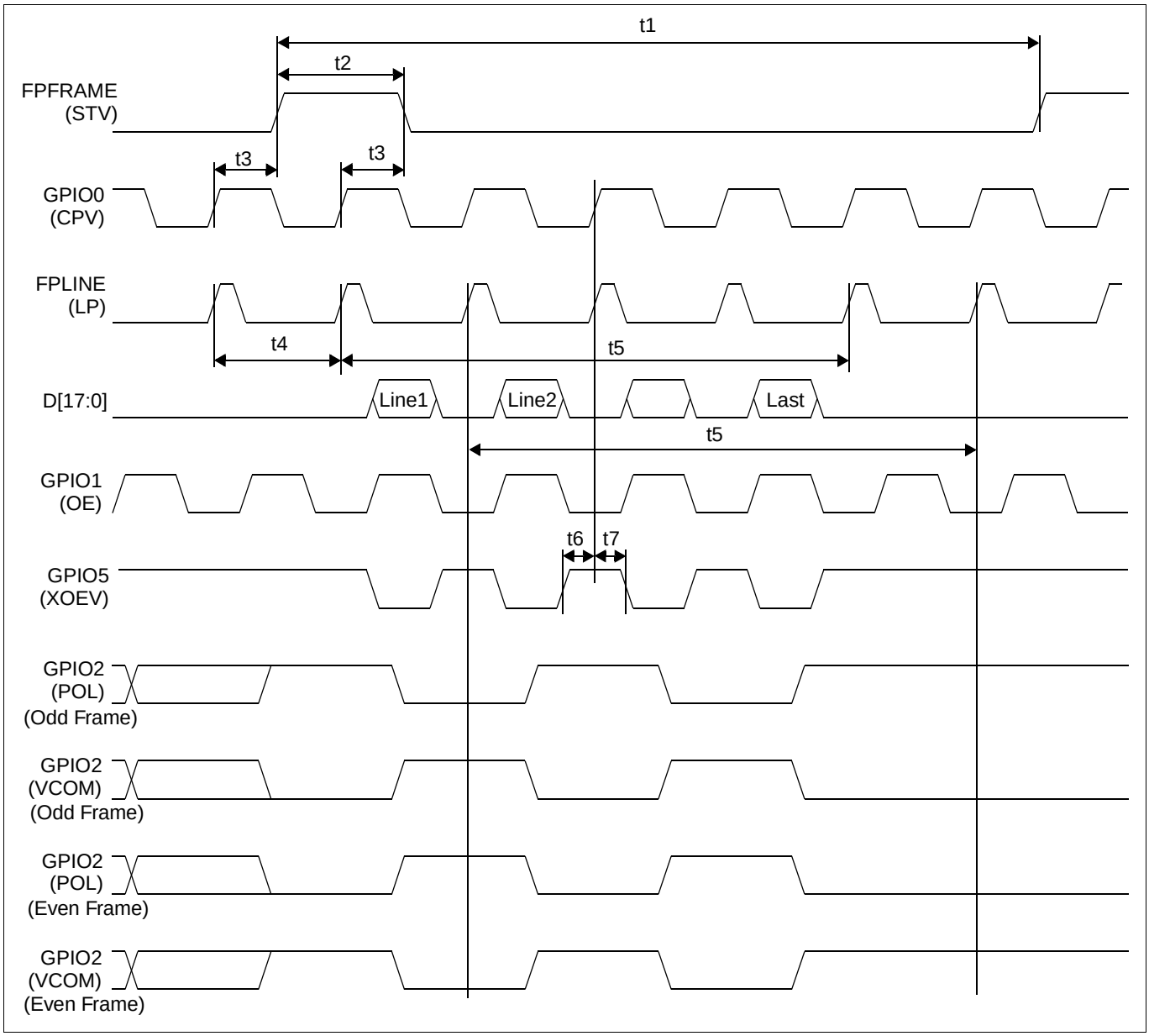


Figure 6-45 TFT Type 3 Vertical Timing

Table 6-39: TFT Type 3 Vertical Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	Vertical total period	1		1024	Lines
t2	FPFRAME pulse width		1		Lines
t3	GPIO0 rising edge to FPFRAME rising (falling) edge		0.5		Lines
t4	Vertical display start position	1	note 2		Lines
t5	Vertical display period	1	note 3	1024	Lines
t6	GPIO5 rising edge to GPIO0 rising edge	0	note 4	512	Ts
t7	GPIO0 rising edge to GPIO5 falling edge	0	note 5	512	Ts

1. Ts = pixel clock period
2. t4typ = (REG[1Fh] bits 1-0, REG[1Eh] bits 7-0)
3. t5typ = (REG[1Dh] bits 1-0, REG[1Ch] bits 7-0) + 1
4. t6typ = ((REG[DAh] bits 7-0) x 2
5. t7typ = ((REG[DBh] bits 7-0) x 2

Note

When REG[DAh] = 0, GPIO5 (XOEV) stays HIGH.

6.4.16 TFT Type 4 Panel Timing

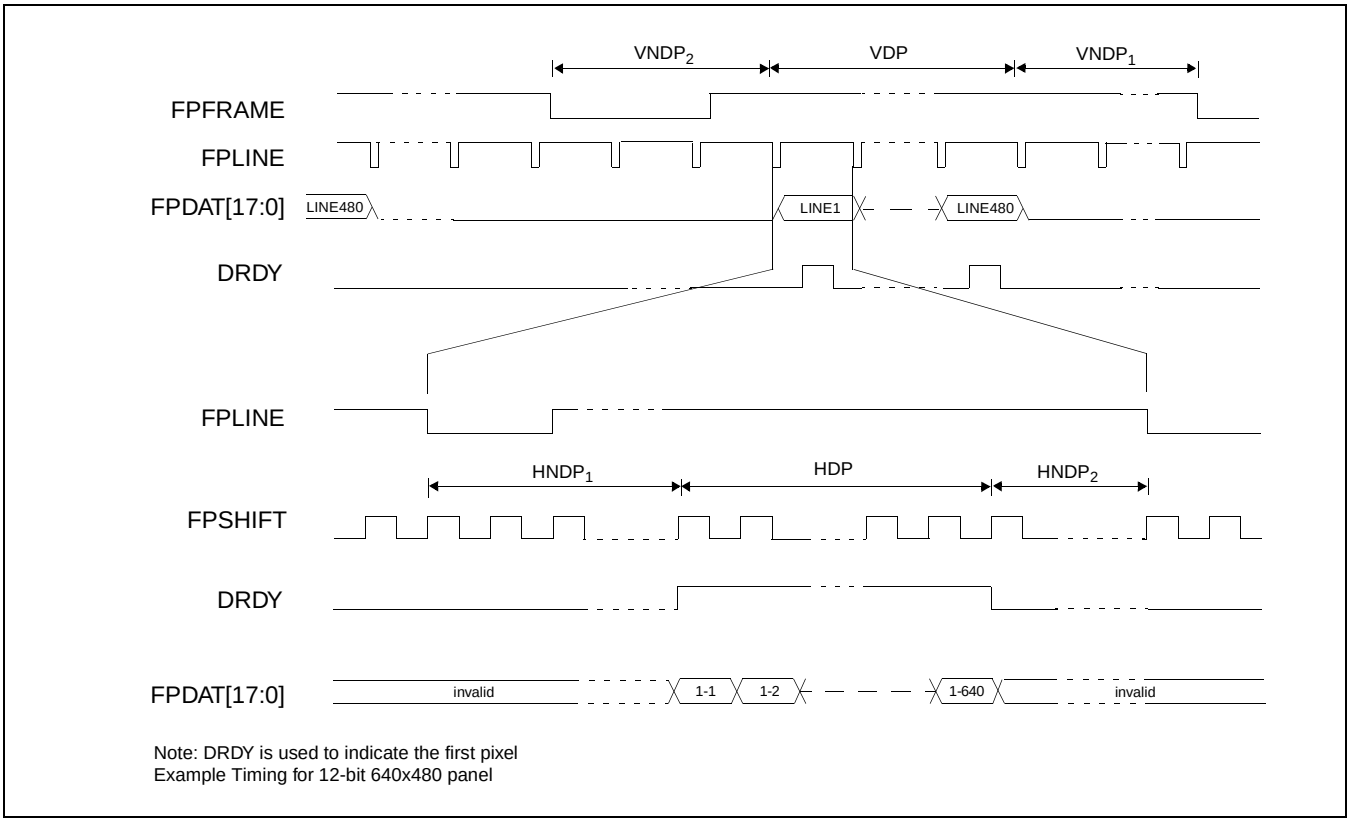


Figure 6-46 TFT Type 4 Panel Timing

VDP = Vertical Display Period
= VDP Lines
VNDP = Vertical Non-Display Period
= VNDP1 + VNDP2
= VT - VDP Lines
VNDP1 = Vertical Non-Display Period 1
= VNDP - VNDP2 Lines
VNDP2 = Vertical Non-Display Period 2
= VDPS - VPS Lines if negative add VT
HDP = Horizontal Display Period
= HDP Ts
HNDP = Horizontal Non-Display Period
= HNDP1 + HNDP2
= HT - HDP Ts
HNDP1 = Horizontal Non-Display Period 1
= HDPS - (HPS + 1) + 5 Ts if negative add HT
HNDP2 = Horizontal Non-Display Period 2
= (HPS + 1) - (HDP + HDPS + 5) Ts if negative add HT

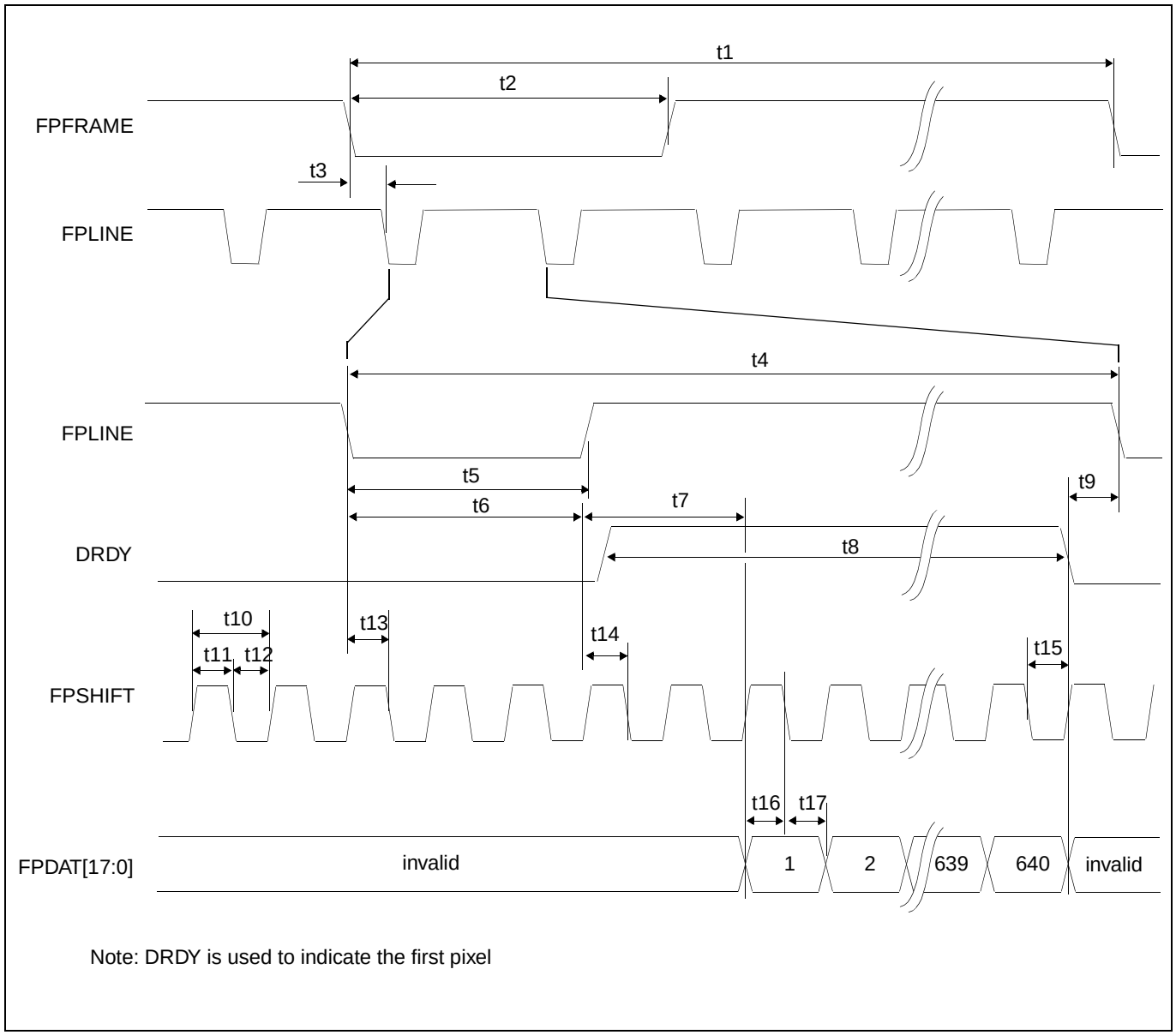


Figure 6-47 TFT Type 4 A.C. Timing

Table 6-40: TFT Type 4 A.C. Timing

Symbol	Parameter	Min	Typ	Max	Units
t1	FPFRAME cycle time	VT	note 2		Lines
t2	FPFRAME pulse width low	VPW	note 3		Lines
t3	FPFRAME falling edge to FPLINE falling edge phase difference	HPS	note 4		Ts (note 1)
t4	FPLINE cycle time	HT	note 5		Ts
t5	FPLINE pulse width low	HPW	note 6		Ts
t6	FPLINE Falling edge to DRDY active		note 7	250	Ts
t7	DRDY active to data setup		8		Ts
t8	DRDY pulse width	HDP	note 8		Ts
t9	DRDY falling edge to FPLINE falling edge		note 9		Ts
t10	FPSHIFT period	1			Ts
t11	FPSHIFT pulse width high	0.5			Ts
t12	FPSHIFT pulse width low	0.5			Ts
t13	FPLINE setup to FPSHIFT falling edge	0.5			Ts
t14	DRDY to FPSHIFT falling edge setup time	0.5			Ts
t15	DRDY hold from FPSHIFT falling edge	0.5			Ts
t16	Data setup to FPSHIFT falling edge	0.5			Ts
t17	Data hold from FPSHIFT falling edge	0.5			Ts

1. Ts = pixel clock period
2. t1typ = (REG[19h] bits 1-0, REG[18h] bits 7-0) +1
3. t2typ = (REG[24h] bits 2-0) + 1
4. t3typ = (REG[23h] bits 1-0, REG[22h] bits 7-0) + 1
5. t4typ = ((REG[12h] bits 6-0) + 1) x 8
6. t5typ = (REG[12h] bits 6-0) + 1
7. t6typ = HDPS- (HPS + 1) x 8
8. t8typ = ((REG[14h] bits 6-0) + 1) x 8
9. t9typ = HPS - HDPS - HDP

7 Clocks

7.1 Clock Descriptions

7.1.1 BCLK

BCLK is an internal clock derived from CLKI or XTAL. CLKI is typically provided from the host CPU bus clock. The source clock options for BCLK may be selected as in the following table.

Table 7-1: BCLK Clock Selection

Source Clock Options	BCLK Selection
CLKI	CNF[7:6] = 00, REG[CAh] bit 0 = 0
CLKI ÷2	CNF[7:6] = 01, REG[CAh] bit 0 = 0
CLKI ÷3[1]	CNF[7:6] = 10, REG[CAh] bit 0 = 0
CLKI ÷4[1]	CNF[7:6] = 11, REG[CAh] bit 0 = 0
XTAL[2]	CNF[7:6] = 00, REG[CAh] bit 0 = 1
XTAL[2] ÷2	CNF[7:6] = 01, REG[CAh] bit 0 = 1
XTAL[2] ÷3[1]	CNF[7:6] = 10, REG[CAh] bit 0 = 1
XTAL[2] ÷4[1]	CNF[7:6] = 11, REG[CAh] bit 0 = 1

Note

[1] The ÷ 3 and ÷ 4 options may not work properly with bus interfaces which have short back-to-back cycle timing.

[2] XTAL should only be used for BCLK when using the Indirect Interface.

7.1.2 MCLK

MCLK provides the internal clock required to access the embedded SRAM. The S1D13708 is designed with efficient power saving control for clocks (clocks are turned off when not used); reducing the frequency of MCLK does not necessarily save more power. Furthermore, reducing the MCLK frequency relative to the BCLK frequency increases the CPU cycle latency and so reduces screen update performance. For a balance of power saving and performance, MCLK should be configured to have a high enough frequency setting to provide sufficient screen refresh as well as acceptable CPU cycle latency.

The source clock options for MCLK may be selected as in the following table.

Table 7-2: MCLK Clock Selection

Source Clock Options	MCLK Selection
BCLK	REG[04h] bit 5,4 = 00
BCLK ÷2	REG[04h] bit 5,4 = 01
BCLK ÷3	REG[04h] bit 5,4 = 10
BCLK ÷4	REG[04h] bit 5,4 = 11

7.1.3 PCLK

PCLK is the internal clock used to control the LCD panel. PCLK should be chosen to match the optimum frame rate of the LCD panel. See Section 9, “Frame Rate Calculation” on page 171 for details on the relationship between PCLK and frame rate.

Some flexibility is possible in the selection of PCLK. Firstly, LCD panels typically have a range of permissible frame rates. Secondly, it may be possible to choose a higher PCLK frequency and tailor the horizontal and vertical non-display periods to lower the frame-rate to its optimal value.

The source clock options for PCLK may be selected as in the following table.

Table 7-3: PCLK Clock Selection

Source Clock Options	PCLK Selection
MCLK	REG[05h] = 00h
MCLK ÷2	REG[05h] = 10h
MCLK ÷3	REG[05h] = 20h
MCLK ÷4	REG[05h] = 30h
MCLK ÷8	REG[05h] = 40h
BCLK	REG[05h] = 01h
BCLK ÷2	REG[05h] = 11h
BCLK ÷3	REG[05h] = 21h
BCLK ÷4	REG[05h] = 31h
BCLK ÷8	REG[05h] = 41h
CLKI	REG[05h] = 02h
CLKI ÷2	REG[05h] = 12h
CLKI ÷3	REG[05h] = 22h
CLKI ÷4	REG[05h] = 32h
CLKI ÷8	REG[05h] = 42h
CLKI2	REG[CAh] bit 1 = 0, REG[05h] = 03h
CLKI2 ÷2	REG[CAh] bit 1 = 0, REG[05h] = 13h
CLKI2 ÷3	REG[CAh] bit 1 = 0, REG[05h] = 23h
CLKI2 ÷4	REG[CAh] bit 1 = 0, REG[05h] = 33h
CLKI2 ÷8	REG[CAh] bit 1 = 0, REG[05h] = 43h

Table 7-3: PCLK Clock Selection (Continued)

Source Clock Options	PCLK Selection
XTAL	REG[CAh] bit 1 = 1, REG[05h] = 03h
XTAL ÷2	REG[CAh] bit 1 = 1, REG[05h] = 13h
XTAL ÷3	REG[CAh] bit 1 = 1, REG[05h] = 23h
XTAL ÷4	REG[CAh] bit 1 = 1, REG[05h] = 33h
XTAL ÷8	REG[CAh] bit 1 = 1, REG[05h] = 43h

There is a relationship between the frequency of MCLK and PCLK that must be maintained.

Table 7-4: Relationship between MCLK and PCLK

SwivelView Orientation	Color Depth (bpp)	MCLK to PCLK Relationship
SwivelView 0° and 180°	16	$f_{MCLK} \geq f_{PCLK}$
	8	$f_{MCLK} \geq f_{PCLK} \div 2$
	4	$f_{MCLK} \geq f_{PCLK} \div 4$
	2	$f_{MCLK} \geq f_{PCLK} \div 8$
	1	$f_{MCLK} \geq f_{PCLK} \div 16$
SwivelView 90° and 270°	16/8/4/2/1	$f_{MCLK} \geq 1.25 f_{PCLK}$

7.1.4 PWMCLK

PWMCLK is the internal clock used by the Pulse Width Modulator for output to the panel.

The source clock options for PWMCLK may be selected as in the following table.

Table 7-5: PWMCLK Clock Selection

Source Clock Options	PWMCLK Selection
CLKI	REG[B1h] bit 0 = 0
CLKI2	REG[CAh] bit 1 = 0, REG[B1h] bit 0 = 1
XTAL	REG[CAh] bit 1 = 1, REG[B1h] bit 0 = 1

For further information on controlling PWMCLK, see Section 8.3.9, “Pulse Width Modulation (PWM) Clock and Contrast Voltage (CV) Pulse Configuration Registers” on page 150.

Note

The S1D13708 provides Pulse Width Modulation output on the pin PWMOUT. PWMOUT can be used to control LCD panels which support PWM control of the back-light inverter.

7.2 Clock Selection

The following diagram provides a logical representation of the S1D13708 internal clocks.

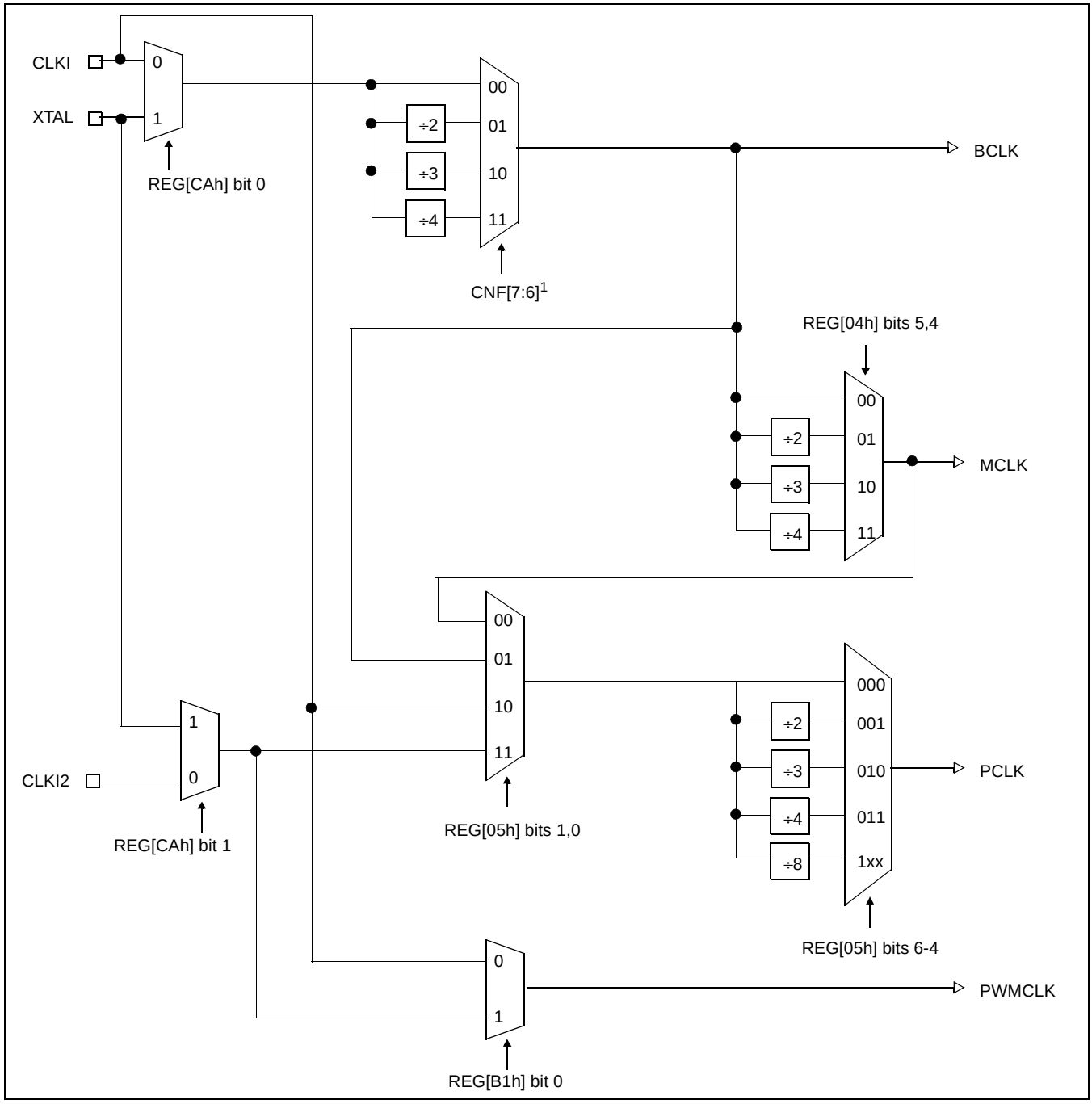


Figure 7-1 Clock Selection

Note

[1] CNF[7:6] must be set at RESET#.

7.3 Clocks versus Functions

Table 7-6: “S1D13708 Internal Clock Requirements”, lists the internal clocks required for the following S1D13708 functions.

Table 7-6: S1D13708 Internal Clock Requirements

Function	Bus Clock (BCLK)	Memory Clock (MCLK)	Pixel Clock (PCLK)	PWM Clock (PWMCLK)
Register Read/Write	Required	Not Required	Not Required	Not Required[1]
Memory Read/Write	Required	Required	Not Required	Not Required[1]
Look-Up Table Register Read/Write	Required	Required	Not Required	Not Required[1]
Software Power Save	Required	Not Required	Not Required	Not Required[1]
LCD Output	Not Required	Required	Required	Not Required[1]

Note
[1]PWMCLK is an optional clock (see Section 7.1.4, “PWMCLK” on page 115).

8 Registers

This section discusses how and where to access the S1D13708 registers. It also provides detailed information about the layout and usage of each register.

8.1 Register Mapping

The S1D13708 registers are memory-mapped. When the system decodes the input pins as CS# = 0 and M/R# = 0, the registers may be accessed. The register space is decoded by A[16:0].

8.2 Register Set

The S1D13708 register set is as follows.

Table 8-1: S1D13708 Register Set

Table 8-1: S1D13708 Register Set

8.3 Register Descriptions

Unless specified otherwise, all register bits are set to 0 during power-on.

8.3.1 Read-Only Configuration Registers

Revision Code Register
REG[00h] Read Only

Product Code Bits 5-0						Revision Code Bits 1-0	
7	6	5	4	3	2	1	0

Note
The S1D13708 returns a value of 34h.

bits 7-2 Product Code
These are read-only bits that indicates the product code. The product code is 001101.

bits 1-0 Revision Code
These are read-only bits that indicates the revision code. The revision code is 00.

Display Buffer Size Register
REG[01h] Read Only

Display Buffer Size Bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 Display Buffer Size Bits [7:0]
This is a read-only register that indicates the size of the SRAM display buffer measured in 4K byte increments. The S1D13708 display buffer is 80K bytes and therefore this register returns a value of 20 (14h).

Value of this register = display buffer size ÷ 4K bytes
= 80K bytes ÷ 4K bytes
= 20 (14h)

Configuration Readback Register
REG[02h] Read Only

CNF7 Status	CNF6 Status	CNF5 Status	CNF4 Status	CNF3 Status	CNF2 Status	CNF1 Status	CNF0 Status
7	6	5	4	3	2	1	0

bits 7-0 CNF[7:0] Status
These read-only status bits return the status of the configuration pins CNF[7:0]. CNF[7:0] are latched at the rising edge of RESET#.

8.3.2 Clock Configuration Registers

Memory Clock Configuration Register
REG[04h] Read/Write

n/a		MCLK Divide Select Bits 1-0		n/a			
7	6	5	4	3	2	1	0

bits 5-4 MCLK Divide Select Bits [1:0]
These bits determine the divide used to generate the Memory Clock (MCLK) from the Bus Clock (BCLK).

Table 8-2: MCLK Divide Selection

MCLK Divide Select Bits	BCLK to MCLK Frequency Ratio
00	1:1
01	2:1
10	3:1
11	4:1

Pixel Clock Configuration Register
REG[05h] Read/Write

n/a	PCLK Divide Select Bits 2-0			n/a		PCLK Source Select Bits 1-0	
7	6	5	4	3	2	1	0

bits 6-4 PCLK Divide Select Bits [1:0]
These bits determine the divide used to generate the Pixel Clock (PCLK) from the Pixel Clock Source.

Table 8-3: PCLK Divide Selection

PCLK Divide Select Bits	PCLK Source to PCLK Frequency Ratio
000	1:1
001	2:1
010	3:1
011	4:1
1XX	8:1

bits 1-0 PCLK Source Select Bits [1:0]
These bits determine the source of the Pixel Clock (PCLK).

Table 8-4: PCLK Source Selection

PCLK Source Select Bits 1:0	PCLK Source
00	MCLK
01	BCLK
10	CLKI
11	CLKI2 / XTAL

Note
Selecting XTAL as the PCLK source is controlled by the BCLK Source Select bit (REG[C9h] bit 0).

8.3.3 Look-Up Table Registers

Look-Up Table Blue Write Data Register
REG[08h] Write Only

LUT Blue Write Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Blue Write Data Bits [5:0]
This register contains the data to be written to the blue component of the Look-Up Table. The data is stored in this register until a write to the LUT Write Address register (REG[0Bh]) moves the data into the Look-Up Table.

Note
The LUT entry is updated only when the LUT Write Address Register (REG[0Bh]) is written to.

Look-Up Table Green Write Data Register
REG[09h] Write Only

LUT Green Write Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Green Write Data Bits [5:0]
This register contains the data to be written to the green component of the Look-Up Table. The data is stored in this register until a write to the LUT Write Address register (REG[0Bh]) moves the data into the Look-Up Table.

Note
The LUT entry is updated only when the LUT Write Address Register (REG[0Bh]) is written to.

Look-Up Table Red Write Data Register
REG[0Ah] Write Only

LUT Red Write Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Red Write Data Bits [5:0]
This register contains the data to be written to the red component of the Look-Up Table. The data is stored in this register until a write to the LUT Write Address register (REG[0Bh]) moves the data into the Look-Up Table.

Note
The LUT entry is updated only when the LUT Write Address Register (REG[0Bh]) is written to.

Look-Up Table Write Address Register
REG[0Bh] Write Only

LUT Write Address Bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 LUT Write Address Bits [7:0]
This register forms a pointer into the Look-Up Table (LUT) which is used to write LUT data stored in REG[08h], REG[09h] and REG[0Ah. The LUT Read Address Register (REG[0Fh]) and all corresponding Read Data Registers will also be updated with the newly written address and data. This is a write-only register and returns 00h if read.Fh

Note
The S1D13708 has three 256-position, 6-bit wide LUTs, one for each of red, green, and blue (see Section 11, “Look-Up Table Architecture” on page 173).

Look-Up Table Blue Read Data Register
REG[0Ch] Read Only

LUT Blue Read Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Blue Read Data Bits [5:0]
This register contains the data from the blue component of the Look-Up Table. The LUT entry read is controlled by the LUT Read Address Register (REG[0Fh]). This is a read-only register.

Look-Up Table Green Read Data Register
REG[0Dh] Read Only

LUT Green Read Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Green Read Data Bits [5:0]
This register contains the data from the green component of the Look-Up Table. The LUT entry read is controlled by the LUT Read Address Register (REG[0Fh]). This is a read-only register.

Look-Up Table Red Read Data Register
REG[0Eh] Read Only

LUT Red Read Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Red Read Data Bits [5:0]
This register contains the data from the red component of the Look-Up Table. The LUT entry read is controlled by the LUT Read Address Register (REG[0Fh]). This is a read-only register.

Look-Up Table Read Address Register
REG[0Fh] Write Only

LUT Read Address Bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 LUT Read Address Bits [7:0]
This register forms a pointer into the Look-Up Table (LUT) which is used to read LUT data and store it in REG[0Ch], REG[0Dh], REG[0Eh]. This is a write-only register and returns 00h if read.

Note
The S1D13708 has three 256-position, 6-bit wide LUTs, one for each of red, green, and blue (see Section 11, “Look-Up Table Architecture” on page 173).

8.3.4 Panel Configuration Registers

Panel Type Register REG[10h]							Read/Write
Panel Data Format Select	Color/Mono. Panel Select	Panel Data Width Bits 1-0		HRTFT/ D-TFD Control Bit	n/a	Panel Type Select Bits 1-0	
7	6	5	4	3	2	1	0

bit 7 Panel Data Format Select
When this bit = 0, 8-bit single color passive LCD panel data format 1 is selected. For AC timing see Section 6.4.5, “Single Color 8-Bit Panel Timing (Format 1)” on page 78.
When this bit = 1, 8-bit single color passive LCD panel data format 2 is selected. For AC timing see Section 6.4.6, “Single Color 8-Bit Panel Timing (Format 2)” on page 80.

bit 6 Color/Mono Panel Select
When this bit = 0, a monochrome LCD panel is selected.
When this bit = 1, a color LCD panel is selected.

bits 5-4 Panel Data Width Bits [1:0]
These bits select the data width size of the LCD panel.

Table 8-5: Panel Data Width Selection

Panel Data Width Bits [1:0]	Passive Panel Data Width Size	Active Panel Data Width Size
00	4-bit	9-bit
01	8-bit	12-bit
10	16-bit	18-bit
11	Reserved	Reserved

bit 3 HRTFT/D-TFD Control Bit
This bit selects one of two modes of operation for HR-TFT/D-TFD panels depending on the panel resolution. This bit has no effect for other panel types.

Table 8-6: HRTFT/D-TFD Panel Resolution Selection

HRTFT/D-TFD Control Bit	HR-TFT Resolution	D-TFD Resolution
0	160x160	160x240
1	320x240	320x240

Note
This bit sets some internal non-configurable timing values for the selected panel. However, all panel configuration registers (REG[12h] - REG[27h]) still require programming with the appropriate values for the selected panel. For panel AC timing, see Section 6.4, “Display Interface” on page 68.

bits 1-0 Panel Type Select Bits[1:0]
These bits select the panel type.

Table 8-7: LCD Panel Type Selection

REG[10h] Bits[1:0]	Panel Type Select
00	STN
01	TFT
10	HR-TFT
11	D-TFD

MOD Rate Register
REG[11h] Read/Write

7	6	5	4	3	2	1	0
n/a		MOD Rate Bits 5-0					

bits 5-0 MOD Rate Bits [5:0]
These bits are for passive LCD panels only.
When these bits are all 0, the MOD output signal (DRDY) toggles every FPFRAME.
For a non-zero value *n*, the MOD output signal (DRDY) toggles every n FPLINE.

Horizontal Total Register
REG[12h] Read/Write

7	6	5	4	3	2	1	0
n/a	Horizontal Total Bits 6-0						

bits 6-0 Horizontal Total Bits [6:0]
These bits specify the LCD panel Horizontal Total period, in 8 pixel resolution. The Horizontal Total is the sum of the Horizontal Display period and the Horizontal Non-Display period. Since the maximum Horizontal Total is 1024 pixels, the maximum panel resolution supported is 800x600.

Horizontal Total in number of pixels = ((REG[12h] bits 6:0) + 1) × 8

Note
For TFT Type 3 panels this register must be programmed such that the following formula is valid.
HT - HDPS - HDP < 11 PCLKs

Note
This register must be programmed such that the following formula is valid.
HDPS + HDP < HT

Note
For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 68.

Horizontal Display Period Register
REG[14h] Read/Write

7	6	5	4	3	2	1	0
n/a	Horizontal Display Period Bits 6-0						

bits 6-0 Horizontal Display Period Bits [6:0]
These bits specify the LCD panel Horizontal Display period, in 8 pixel resolution. The Horizontal Display period should be less than the Horizontal Total to allow for a sufficient Horizontal Non-Display period.

Horizontal Display Period in number of pixels = ((REG[14h] bits 6:0) + 1) × 8

Note
For passive panels, HDP must be a minimum of 32 pixels and can be increased by multiples of 16. For TFT panels, HDP must be a minimum of 16 pixels and can be increased by multiples of 8.

Note
For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 68.

Horizontal Display Period Start Position Register 0
REG[16h] Read/Write

7	6	5	4	3	2	1	0
Horizontal Display Period Start Position Bits 7-0							

Horizontal Display Period Start Position Register 1
REG[17h] Read/Write

7	6	5	4	3	2	1	0
n/a						Horizontal Display Period Start Position Bits 9-8	

bits 9-0 Horizontal Display Period Start Position Bits [9:0]
These bits specify a value used in the calculation of the Horizontal Display Period Start Position (in 1 pixel resolution) for TFT, ‘Direct’ HR-TFT and ‘Direct’ D-TFD panels. For passive LCD panels these bits must be set to 00h.

To calculate the Horizontal Display Period Start Position (HDPS) an offset which depends on the panel type is required. For further information on calculating the HDPS, see the specific panel AC timing in Section 6.4, “Display Interface” on page 68.

Note
This register must be programmed such that the following formula is valid.
HDPS + HDP < HT

Vertical Total Register 0
REG[18h] Read/Write

Vertical Total Bits 7-0							
7	6	5	4	3	2	1	0

Vertical Total Register 1
REG[19h] Read/Write

n/a						Vertical Total Bits 9-8	
7	6	5	4	3	2	1	0

bits 9-0 Vertical Total Bits [9:0]
These bits specify the LCD panel Vertical Total period, in 1 line resolution. The Vertical Total is the sum of the Vertical Display Period and the Vertical Non-Display Period. The maximum Vertical Total is 1024 lines.

Vertical Total in number of lines = (REG[18h] bits 7:0, REG[19h] bits 1:0) + 1

Note
[1] This register must be programmed such that the following formula is valid.
VDPS + VDP < VT
[2] For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 68.

Vertical Display Period Register 0
REG[1Ch] Read/Write

Vertical Display Period Bits 7-0							
7	6	5	4	3	2	1	0

Vertical Display Period Register 1
REG[1Dh] Read/Write

n/a						Vertical Display Period Bits 9-8	
7	6	5	4	3	2	1	0

bits 9-0 Vertical Display Period Bits [9:0]
These bits specify the LCD panel Vertical Display period, in 1 line resolution. The Vertical Display period should be less than the Vertical Total to allow for a sufficient Vertical Non-Display period.

Vertical Display Period in number of lines = (REG[1Ch] bits 7:0, REG[1Dh] bits 1:0) + 1

Note
For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 68.

Vertical Display Period Start Position Register 0
REG[1Eh] Read/Write

7	6	5	4	3	2	1	0
Vertical Display Period Start Position Bits 7-0							

Vertical Display Period Start Position Register 1
REG[1Fh] Read/Write

7	6	5	4	3	2	1	0
n/a						Vertical Display Period Start Position Bits 9-8	

bits 9-0 Vertical Display Period Start Position Bits [9:0]
These bits specify the Vertical Display Period Start Position for HR-TFT and D-TFD panels in 1 line resolution.

Note
For passive LCD and TFT (non-HR-TFT/D-TFD) panels these bits must be set to 00h.

Note
This register must be programmed such that the following formula is valid.
VDPS + VDP < VT

Note
For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 68.

FPLINE Pulse Width Register
REG[20h] Read/Write

7	6	5	4	3	2	1	0
FPLINE Pulse Polarity	FPLINE Pulse Width Bits 6-0						

bit 7 FPLINE Pulse Polarity
This bit selects the polarity of the horizontal sync signal. For most passive panels this bit should be set to 1. For TFT panels this bit is set according to the horizontal sync signal required by the panel, typically FPLINE or LP, depending on the panel type.
When this bit = 0, the horizontal sync signal is active low.
When this bit = 1, the horizontal sync signal is active high.

bits 6-0 FPLINE Pulse Width Bits [6:0]
These bits specify the width of the panel horizontal sync signal, in 1 pixel resolution. The horizontal sync signal is typically FPLINE or LP, depending on the panel type.

FPLINE Pulse Width in number of pixels = (REG[20h] bits 6:0) + 1

Note
For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 68.

FPLINE Pulse Start Position Register 0
REG[22h] Read/Write

FPLINE Pulse Start Position Bits 7-0							
7	6	5	4	3	2	1	0

FPLINE Pulse Start Position Register 1
REG[23h] Read/Write

n/a						FPLINE Pulse Start Position Bits 9-8	
7	6	5	4	3	2	1	0

bits 9-0 FPLINE Pulse Start Position Bits [9:0]
These bits specify the start position of the horizontal sync signal, in 1 pixel resolution.

FPLINE Pulse Start Position in pixels = [(REG[23h] bits 1-0, REG[22h] bits 7-0) + 1]

Note
For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 68.

FPFRAME Pulse Width Register
REG[24h] Read/Write

FPFRAME Pulse Polarity	n/a				FPFRAME Pulse Width Bits 2-0		
7	6	5	4	3	2	1	0

bit 7 FPFRAME Pulse Polarity
This bit selects the polarity of the vertical sync signal. For most passive panels this bit should be set to 1. For TFT panels this bit is set according to the vertical sync signal required by the panel, typically FPFRAME, SPS or DY, depending on the panel type.
When this bit = 0, the vertical sync signal is active low.
When this bit = 1, the vertical sync signal is active high.

bits 2-0 FPFRAME Pulse Width Bits [2:0]
These bits specify the width of the panel vertical sync signal, in 1 line resolution. The vertical sync signal is typically FPFRAME, SPS or DY, depending on the panel type.

FPFRAME Pulse Width in number of lines = (REG[24h] bits 2:0) + 1

Note
For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 68.

FPFRAME Pulse Start Position Register 0
REG[26h] Read/Write

7	6	5	4	3	2	1	0
FPFRAME Pulse Start Position Bits 7-0							

FPFRAME Pulse Start Position Register 1
REG[27h] Read/Write

7	6	5	4	3	2	1	0
n/a						FPFRAME Pulse Start Position Bits 9-8	

bits 9-0 FPFRAME Pulse Start Position Bits [9:0]
These bits specify the start position of the vertical sync signal, in 1 line resolution.

Note
For panel AC timing and timing parameter definitions, see Section 6.4, “Display Interface” on page 68.

D-TFD GCP Index Register
REG[28h] Read/Write

7	6	5	4	3	2	1	0
n/a			D-TFD GCP Index Bits 4-0				

bits 4-0 D-TFD GCP Index Bits [4:0]
For D-TFD panels only. These bits form the index that points to 32 8-bit GCP data registers.

D-TFD GCP Data Register
REG[2Ch] Read/Write

7	6	5	4	3	2	1	0
D-TFD GCP Data Bits 7-0							

bits 7-0 D-TFD GCP Data Bits [7:0]
For D-TFD panel only. This register stores the data to be written to the GCP data bits and is controlled by the D-TFD GCP Index register (REG[28h]).

Note
The Panel Type bits (REG[10h] bits 1:0) must be set to 11 (D-TFD) for the GCP Data bits to have any hardware effect.

8.3.5 Display Mode Registers

Display Mode Register REG[70h]							Read/Write
Display Blank	Dithering Disable	Hardware Video Invert Enable	Software Video Invert	n/a	Bit-per-pixel Select Bits 2-0		
7	6	5	4	3	2	1	0

bit 7 Display Blank

When this bit = 0, the LCD display pipeline is enabled.
When this bit = 1, the LCD display pipeline is disabled and all LCD data outputs are forced to zero (i.e., the screen is blanked).

bit 6 Dithering Disable

When this bit = 0, dithering on the passive LCD panel is enabled, allowing a maximum of 256K colors (2^{18}) or 64 gray shades in 1/2/4/8 bpp mode. In 16bpp mode, only 64K colors (2^{16}) is allowed because the LUT is bypassed.
When this bit = 1, dithering on the passive LCD panel is disabled, allowing a maximum of 4096 colors (2^{12}) or 16 gray shades.

Note

For a summary of the results of dithering for each color depth, see Table 8-9: "LCD Bit-per-pixel Selection," on page 135.

All passive STN color panels are controlled using 3 bits for each pixel (RGB) for a total of 8 possible colors. LCD controllers use a combination of Frame Rate Modulation (FRM) and dithering to achieve more than 8 colors per pixel. FRM can achieve 16 shades of color for each RGB component resulting in a total of 4096 possible colors (16x16x16). Dithering uses a 4 pixel square formation and applies a set of 4 hard-coded patterns for each of the 16 shades of color. This expands the original 16 shades of color from the FRM logic to 64 shades per RGB component which results in 256K colors per pixel (64x64x64).

For the S1D13708, 16 bpp is arranged as 5-6-5 RGB. In this mode, when dithering is enabled, the LUT is bypassed and the original 16-bit data is used as a pointer into the 64 shades per color in the following manner.

(5-6-5 RGB) 32 possible Red, 64 possible Green, 32 possible Blue

This combination of FRM and dithering results in 256K colors/pixel, however, the 16 bpp limitation of the S1D13708 limits this to 64K colors/pixel.

bit 5 Hardware Video Invert Enable

This bit allows the Video Invert feature to be controlled using the General Purpose IO pin GPIO0. **This option is not available if configured for a HR-TFT, D-TFD, TFT Type 2, or TFT Type 3 as GPIO0 is used as an LCD control signal by these panels.**
When this bit = 0, GPIO0 has no effect on the video data.
When this bit = 1, video data may be inverted via GPIO0.

Note
The S1D13708 requires some configuration before the hardware video invert feature can be enabled.
- CNF3 must be set to 1 at RESET#
- GPIO Pin Input Enable (REG[A9h] bit 7) must be set to 1
- GPIO0 Pin IO Configuration (REG[A8h] bit 0) must be set to 0

If Hardware Video Invert is not available, the video invert function can be controlled by software using REG[70h] bit 4. The following table summarizes the video invert options available.

Table 8-8: Inverse Video Mode Select Options

Hardware Video Invert Enable	Software Video Invert	GPIO0	Video Data
0	0	X	Normal
0	1	X	Inverse
1	X	0	Normal
1	X	1	Inverse

Note
Video data is inverted after the Look-Up Table.

bit 4 Software Video Invert

When this bit = 0, video data is normal.
When this bit = 1, video data is inverted.
See Table 8-8: “Inverse Video Mode Select Options”.

Note
Video data is inverted after the Look-Up Table

bits 2-0 Bit-per-pixel Select Bits [2:0]

These bits select the color depth (bit-per-pixel) for the displayed data for both the main window and the PIP+ window (if active).

1, 2, 4 and 8 bpp modes use the 18-bit LUT, allowing maximum 256K colors. 16 bpp mode bypasses the LUT, allowing only 64K colors.

Table 8-9: LCD Bit-per-pixel Selection

Bit-per-pixel Select Bits [2:0]	Color Depth (bpp)	Maximum Number of Colors/Shades		Max. No. Of Simultaneously Displayed Colors/Shades
		Passive Panel (Dithering On)	TFT Panel	
000	1 bpp	256K/64	256K/64	2/2
001	2 bpp	256K/64	256K/64	4/4
010	4 bpp	256K/64	256K/64	16/16
011	8 bpp	256K/64	256K/64	256/64
100	16 bpp	64K/64	64K/64	64K/64
101, 110, 111	Reserved	n/a	n/a	n/a

Special Effects Register
REG[71h] Read/Write

7	6	5	4	3	2	1	0
Display Data Word Swap	Display Data Byte Swap	n/a	PIP+ Window Enable	n/a		SwivelView Mode Select Bits 1-0	

bit 7 Display Data Word Swap

The display pipe fetches 32-bits of data from the display buffer. This bit enables the lower 16-bit word and the upper 16-bit word to be swapped before sending them to the LCD display. If the Display Data Byte Swap bit is also enabled, then the byte order of the fetched 32-bit data is reversed.

Note

For further information on byte swapping for Big Endian mode, see Section 17, “Big-Endian Bus Interface” on page 212.

bit 6 Display Data Byte Swap

The display pipe fetches 32-bits of data from the display buffer. This bit enables byte 0 and byte 1 to be swapped, and byte 2 and byte 3 to be swapped, before sending them to the LCD display. If the Display Data Word Swap bit is also enabled, then the byte order of the fetched 32-bit data is reversed.

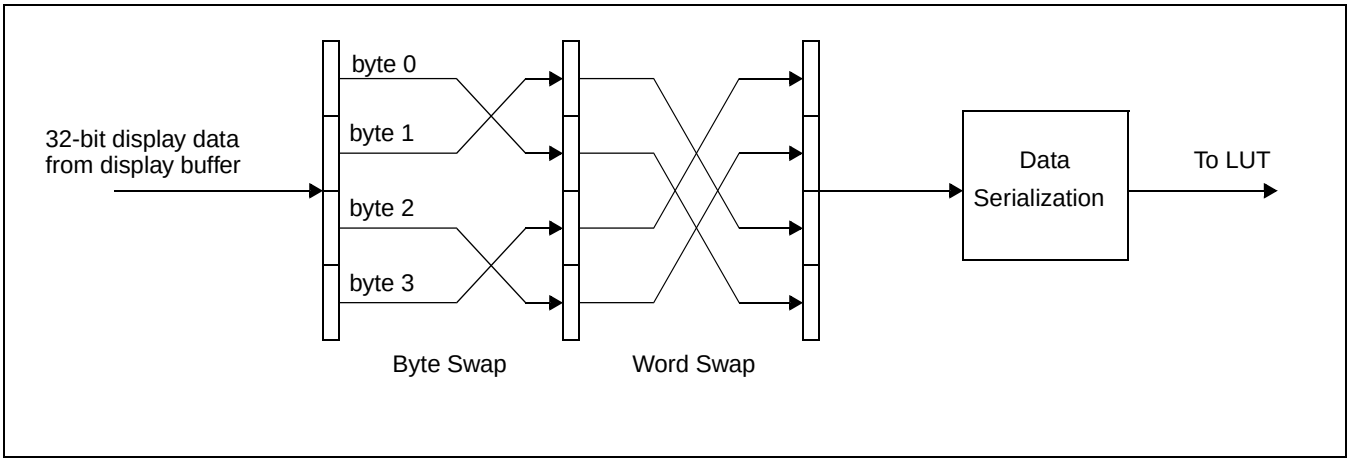


Figure 8-1 Display Data Byte/Word Swap

Note

For further information on byte swapping for Big Endian mode, see Section 17, “Big-Endian Bus Interface” on page 212.

bit 4 PIP+ Window Enable

This bit enables the PIP+ window within the main window used for the Picture-in-Picture Plus feature. The location of the PIP+ window within the landscape window is determined by the PIP+ Window X Position registers (REG[84h], REG[85h], REG[8Ch], REG[8Dh]) and PIP+ Window Y Position registers (REG[88h], REG[89h], REG[90h], REG[91h]). The PIP+ window has its own Display Start Address register (REG[7Ch, REG[7Dh], REG[7Eh]) and Memory Address Offset register (REG[80h], REG[81h]). The PIP+ window shares the same color depth and SwivelView™ orientation as the main window.

Note

Picture-in-Picture Plus (PIP+) is not available when the Ink Layer is enabled (REG[C9h] bit 0 = 1).

bits 1-0 SwivelView Mode Select Bits [1:0]

These bits select different SwivelView™ orientations:

Table 8-10: SwivelView™ Mode Select Options

SwivelView Mode Select Bits	SwivelView Orientation
00	0° (Normal)
01	90°
10	180°
11	270°

Main Window Display Start Address Register 0
REG[74h] Read/Write

Main window Display Start Address Bits 7-0							
7	6	5	4	3	2	1	0

Main Window Display Start Address Register 1
REG[75h] Read/Write

Main window Display Start Address Bits 15-8							
7	6	5	4	3	2	1	0

Main Window Display Start Address Register 2
REG[76h] Read/Write

n/a							Main window Display Start Address Bit 16
7	6	5	4	3	2	1	0

bits 16-0 Main Window Display Start Address Bits [16:0]
These bits form the 17-bit address for the starting double-word of the LCD image in the display buffer for the main window.

Note that this is a double-word (32-bit) address. An entry of 00000h into these registers represents the first double-word of display memory, an entry of 00001h represents the second double-word of the display memory, and so on.

Main Window Line Address Offset Register 0
REG[78h] Read/Write

Main window Line Address Offset Bits 7-0							
7	6	5	4	3	2	1	0

Main Window Line Address Offset Register 1
REG[79h] Read/Write

n/a						Main window Line Address Offset Bits 9-8	
7	6	5	4	3	2	1	0

bits 9-0 Main Window Line Address Offset Bits [9:0]
These bits are the LCD display's 10-bit address offset from the starting double-word of line "n" to the starting double-word of line "n + 1" for the main window. **Note that this is a 32-bit address increment.**

A virtual image can be formed by setting this register to a value greater than the width of the main window. The displayed image is a window into the larger virtual image.

8.3.6 Picture-in-Picture Plus Registers

PIP+ Window Display Start Address Register 0
REG[7C] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window Display Start Address Bits 7-0							

PIP+ Window Display Start Address Register 1
REG[7Dh] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window Display Start Address Bits 15-8							

PIP+ Window Display Start Address Register 2
REG[7Eh] Read/Write

7	6	5	4	3	2	1	0
n/a							PIP+ Window Display Start Address Bit 16

bits 16-0 PIP+ Window Display Start Address Bits [16:0]
These bits form the 17-bit address for the starting double-word of the PIP+ window.

Note that this is a double-word (32-bit) address. An entry of 00000h into these registers represents the first double-word of display memory, an entry of 00001h represents the second double-word of the display memory, and so on.

Note
These bits have no effect unless the Picture-in-Picture Plus PIP+ Window Enable bit is set to 1 (REG[71h] bit 4).

PIP+ Window Line Address Offset Register 0
REG[80h] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window Line Address Offset Bits 7-0							

PIP+ Window Line Address Offset Register 1
REG[81h] Read/Write

7	6	5	4	3	2	1	0
n/a						PIP+ Window Line Address Offset Bits 9-8	

bits 9-0 PIP+ Window Line Address Offset Bits [9:0]
These bits are the LCD display's 10-bit address offset from the starting double-word of line "n" to the starting double-word of line "n + 1" for the PIP+ window. **Note that this is a 32-bit address increment.**

Note
These bits have no effect unless the Picture-in-Picture Plus PIP+ Window Enable bit is set to 1 (REG[71h] bit 4).

PIP+ Window X Start Position Register 0
REG[84h] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window X Start Position Bits 7-0							

PIP+ Window X Start Position Register 1
REG[85h] Read/Write

7	6	5	4	3	2	1	0
n/a						PIP+ Window X Start Position Bits 9-8	

bits 9-0 PIP+ Window X Start Position Bits [9:0]

These bits determine the X start position of the PIP+ window in relation to the origin of the panel. Due to the S1D13708 SwivelView feature, the X start position may not be a horizontal position value (only true in 0° and 180° SwivelView). For further information on defining the value of the X Start Position register, see Section 13, “Picture-in-Picture Plus (PIP+)” on page 184.

The register is also incremented differently based on the SwivelView orientation. For 0° and 180° SwivelView the X start position is incremented by *x* pixels where *x* is relative to the current color depth.

Table 8-11: 32-bit Address Increments for Color Depth

Color Depth	Pixel Increment (x)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

For 90° and 270° SwivelView the X start position is incremented in 1 line increments.

Depending on the color depth, some of the higher bits in this register are unused because the maximum horizontal display width is 1024 pixels.

Note
These bits have no effect unless the Picture-in-Picture Plus PIP+ Window Enable bit is set to 1 (REG[71h] bit 4).

Note
The effect of REG[84h] through REG[91h] takes place only after REG[91h] is written and at the next vertical non-display period.

PIP+ Window Y Start Position Register 0
REG[88h] Read/Write

PIP+ Window Y Start Position Bits 7-0							
7	6	5	4	3	2	1	0

PIP+ Window Y Start Position Register 1
REG[89h] Read/Write

n/a						PIP+ Window Y Start Position Bits 9-8	
7	6	5	4	3	2	1	0

bits 9-0 PIP+ Window Y Start Position Bits [9:0]

These bits determine the Y start position of the PIP+ window in relation to the origin of the panel. Due to the S1D13708 SwivelView feature, the Y start position may not be a vertical position value (only true in 0° and 180° SwivelView). For further information on defining the value of the Y Start Position register, see Section 13, “Picture-in-Picture Plus (PIP+)” on page 184.

The register is also incremented differently based on the SwivelView orientation. For 0° and 180° SwivelView the Y start position is incremented in 1 line increments. For 90° and 270° SwivelView the Y start position is incremented by *y* pixels where *y* is relative to the current color depth.

Table 8-12: 32-bit Address Increments for Color Depth

Color Depth	Pixel Increment (y)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

Depending on the color depth, some of the higher bits in this register are unused because the maximum vertical display height is 1024 pixels.

Note
These bits have no effect unless the Picture-in-Picture Plus PIP+ Window Enable bit is set to 1 (REG[71h] bit 4).

Note
The effect of REG[84h] through REG[91h] takes place only after REG[91h] is written and at the next vertical non-display period.

PIP+ Window X End Position Register 0
REG[8Ch] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window X End Position Bits 7-0							

PIP+ Window X End Position Register 1
REG[8Dh] Read/Write

7	6	5	4	3	2	1	0
n/a						PIP+ Window X End Position Bits 9-8	

bits 9-0 PIP+ Window X End Position Bits [9:0]

These bits determine the X end position of the PIP+ window in relation to the origin of the panel. Due to the S1D13708 SwivelView feature, the X end position may not be a horizontal position value (only true in 0° and 180° SwivelView). For further information on defining the value of the X End Position register, see Section 13, “Picture-in-Picture Plus (PIP+)” on page 184.

The register is also incremented differently based on the SwivelView orientation. For 0° and 180° SwivelView the X end position is incremented by *x* pixels where *x* is relative to the current color depth.

Table 8-13: 32-bit Address Increments for Color Depth

Color Depth	Pixel Increment (x)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

For 90° and 270° SwivelView the X end position is incremented in 1 line increments.

Depending on the color depth, some of the higher bits in this register are unused because the maximum horizontal display width is 1024 pixels.

Note
These bits have no effect unless the Picture-in-Picture Plus PIP+ Window Enable bit is set to 1 (REG[71h] bit 4).

Note
The effect of REG[84h] through REG[91h] takes place only after REG[91h] is written and at the next vertical non-display period.

PIP+ Window Y End Position Register 0
REG[90h] Read/Write

PIP+ Window Y End Position Bits 7-0							
7	6	5	4	3	2	1	0

PIP+ Window Y End Position Register 1
REG[91h] Read/Write

n/a						PIP+ Window Y End Position Bits 9-8	
7	6	5	4	3	2	1	0

bits 9-0 PIP+ Window Y End Position Bits [9:0]

These bits determine the Y end position of the PIP+ window in relation to the origin of the panel. Due to the S1D13708 SwivelView feature, the Y end position may not be a vertical position value (only true in 0° and 180° SwivelView). For further information on defining the value of the Y End Position register, see Section 13, “Picture-in-Picture Plus (PIP+)” on page 184.

The register is also incremented differently based on the SwivelView orientation. For 0° and 180° SwivelView the Y end position is incremented in 1 line increments. For 90° and 270° SwivelView the Y end position is incremented by *y* pixels where *y* is relative to the current color depth.

Table 8-14: 32-bit Address Increments for Color Depth

Color Depth	Pixel Increment (y)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

Depending on the color depth, some of the higher bits in this register are unused because the maximum vertical display height is 1024 pixels.

Note
These bits have no effect unless the Picture-in-Picture Plus PIP+ Window Enable bit is set to 1 (REG[71h] bit 4).

Note
The effect of REG[84h] through REG[91h] takes place only after REG[91h] is written and at the next vertical non-display period.

8.3.7 Miscellaneous Registers

Power Save Configuration Register
REG[A0h] Read/Write

7	6	5	4	3	2	1	0
Vertical Non-Display Period Status (RO)	n/a			Memory Controller Power Save Status (RO)	n/a		Power Save Mode Enable

bit 7 Vertical Non-Display Period Status
This is a read-only status bit.
When this bit = 0, the LCD panel output is in a Vertical Display Period.
When this bit = 1, the LCD panel output is in a Vertical Non-Display Period.

bit 3 Memory Controller Power Save Status
This read-only status bit indicates the power save state of the memory controller.
When this bit = 0, the memory controller is powered up.
When this bit = 1, the memory controller is powered down and the MCLK source can be turned off.

Note
Memory writes are possible during power save mode because the S1D13708 dynamically enables the memory controller for display buffer writes.

bit 0 Power Save Mode Enable
When this bit = 1, the software initiated power save mode is enabled.
When this bit = 0, the software initiated power save mode is disabled.
At reset, this bit is set to 1. For a summary of Power Save Mode, see Section 18, “Power Save Mode” on page 215.

Note
Memory writes are possible during power save mode because the S1D13708 dynamically enables the memory controller for display buffer writes.

Reserved
REG[A1h] Read/Write

7	6	5	4	3	2	1	0
n/a							Reserved

bit 0 Reserved.
This bit must be set to 0.

Reserved
REG[A2h] Read/Write

7	6	5	4	3	2	1	0
Reserved	n/a						Reserved

bit 7 Reserved.
This bit must be set to 0.

bit 0 Reserved.
This bit must be set to 0.

Reserved
REG[A3h] Read/Write

7	6	5	4	3	2	1	0
Reserved	n/a						

bit 7 Reserved.
This bit must be set to 0.

Scratch Pad Register 0
REG[A4h] Read/Write

7	6	5	4	3	2	1	0
Scratch Pad Bits 7-0							

Scratch Pad Register 1
REG[A5h] Read/Write

7	6	5	4	3	2	1	0
Scratch Pad Bits 15-8							

bits 15-0 Scratch Pad Bits [15:0]
This register contains general purpose read/write bits. These bits have no effect on hardware.

8.3.8 General IO Pins Registers

General Purpose IO Pins Configuration Register 0 REG[A8h]							Read/Write
n/a	GPIO6 Pin IO Configuration	GPIO5 Pin IO Configuration	GPIO4 Pin IO Configuration	GPIO3 Pin IO Configuration	GPIO2 Pin IO Configuration	GPIO1 Pin IO Configuration	GPIO0 Pin IO Configuration
7	6	5	4	3	2	1	0

Note
If CNF3 = 0 at RESET#, then all GPIO pins are configured as outputs only and this register has no effect. This case allows the GPIO pins to be used by the HR-TFT/D-TFD panel interfaces. For a summary of GPIO usage for HR-TFT/D-TFD, see Table 4-10: "LCD Interface Pin Mapping," on page 40.

Note
The input functions of the GPIO pins are not enabled until REG[A9h] bit 7 is set to 1.

bit 6 GPIO6 Pin IO Configuration
When this bit = 0 (default), GPIO6 is configured as an input pin.
When this bit = 1, GPIO6 is configured as an output pin.

bit 5 GPIO5 Pin IO Configuration
When this bit = 0 (default), GPIO5 is configured as an input pin.
When this bit = 1, GPIO5 is configured as an output pin.

bit 4 GPIO4 Pin IO Configuration
When this bit = 0 (default), GPIO4 is configured as an input pin.
When this bit = 1, GPIO4 is configured as an output pin.

bit 3 GPIO3 Pin IO Configuration
When this bit = 0 (default), GPIO3 is configured as an input pin.
When this bit = 1, GPIO3 is configured as an output pin.

bit 2 GPIO2 Pin IO Configuration
When this bit = 0 (default), GPIO2 is configured as an input pin.
When this bit = 1, GPIO2 is configured as an output pin.

bit 1 GPIO1 Pin IO Configuration
When this bit = 0 (default), GPIO1 is configured as an input pin.
When this bit = 1, GPIO1 is configured as an output pin.

bit 0 GPIO0 Pin IO Configuration
When this bit = 0 (default), GPIO0 is configured as an input pin.
When this bit = 1, GPIO0 is configured as an output pin.

General Purpose IO Pins Configuration Register 1
REG[A9h] Read/Write

GPIO Pin Input Enable	n/a						
7	6	5	4	3	2	1	0

bit 7 GPIO Pin Input Enable
This bit is used to enable the input function of the GPIO pins. It must be changed to a 1 after power-on reset to enable the input function of the GPIO pins (default is 0).

General Purpose IO Pins Status/Control Register 0
REG[ACh] Read/Write

n/a	GPIO6 Pin IO Status	GPIO5 Pin IO Status	GPIO4 Pin IO Status	GPIO3 Pin IO Status	GPIO2 Pin IO Status	GPIO1 Pin IO Status	GPIO0 Pin IO Status
7	6	5	4	3	2	1	0

Note
For information on GPIO pin mapping when HR-TFT/D-TFD panels are selected, see Table 4-10: “LCD Interface Pin Mapping,” on page 40.

bit 6 GPIO6 Pin IO Status
When GPIO6 is not used as a LCD signal and GPIO6 is configured as an output, writing a 1 to this bit drives GPIO6 high and writing a 0 to this bit drives GPIO6 low.
When GPIO6 is not used as a LCD signal and GPIO6 is configured as an input, a read from this bit returns the status of GPIO6.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) writing to this bit has no effect.

When a TFT Type 3 panel is enabled (REG[C5h] bits 1:0 = 10) writing to this bit has no effect.

bit 5 GPIO5 Pin IO Status
When GPIO5 is not used as a LCD signal and GPIO5 is configured as an output, writing a 1 to this bit drives GPIO5 high and writing a 0 to this bit drives GPIO5 low.
When GPIO5 is not used as a LCD signal and GPIO5 is configured as an input, a read from this bit returns the status of GPIO5.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 1 is written to this bit, the D-TFD signal DD_P1 signal is enabled.
When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 0 is written to this bit, the D-TFD signal DD_P1 signal is forced low.

When a TFT Type 3 panel is enabled (REG[C5h] bits 1:0 = 10) and a 1 is written to this bit, the GPIO5 (XOEV) signal is enabled.
When a TFT Type 3 panel is enabled (REG[C5h] bits 1:0 = 10) and a 0 is written to this bit, the GPIO5 (XOEV) signal is forced high when the FPFRAME Polarity bit = 1 (REG[24h] bit 7 = 1) and forced low when the FPFRAME Polarity bit = 0 (REG[24h] bit 7 = 0).

bit 4 GPIO4 Pin IO Status

When GPIO4 is not used as a LCD signal and GPIO4 is configured as an output, writing a 1 to this bit drives GPIO4 high and writing a 0 to this bit drives GPIO4 low.
When GPIO4 is not used as a LCD signal and GPIO4 is configured as an input, a read from this bit returns the status of GPIO4.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 1 is written to this bit, the D-TFD signal RES signal is enabled.
When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 0 is written to this bit, the D-TFD signal RES signal is forced low.

When a TFT Type 3 panel is enabled (REG[C5h] bits 1:0 = 10) and a 1 is written to this bit, the GPIO4 (VCOM) signal is enabled.
When a TFT Type 3 panel is enabled (REG[C5h] bits 1:0 = 10) and a 0 is written to this bit, the GPIO4 (VCOM) signal is forced low.

bit 3 GPIO3 Pin IO Status

When GPIO3 is not used as a LCD signal and GPIO3 is configured as an output, writing a 1 to this bit drives GPIO3 high and writing a 0 to this bit drives GPIO3 low.
When GPIO3 is not used as a LCD signal and GPIO3 is configured as an input, a read from this bit returns the status of GPIO3.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) writing to this bit has no effect.

When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) writing to this bit has no effect.

When a TFT Type 2 panel is enabled (REG[C5h] bits 1:0 = 01) writing to this bit has no effect.

When a TFT Type 3 panel is enabled (REG[C5h] bits 1:0 = 10) writing to this bit has no effect.

bit 2 GPIO2 Pin IO Status

When GPIO2 is not used as a LCD signal and GPIO2 is configured as an output, writing a 1 to this bit drives GPIO2 high and writing a 0 to this bit drives GPIO2 low.
When GPIO2 is not used as a LCD signal and GPIO2 is configured as an input, a read from this bit returns the status of GPIO2.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) writing to this bit has no effect.

When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) writing to this bit has no effect.

When a TFT Type 2 panel is enabled (REG[C5h] bits 1:0 = 01) writing to this bit has no effect.

When a TFT Type 3 panel is enabled (REG[C5h] bits 1:0 = 10) writing to this bit has no effect.

bit 1 GPIO1 Pin IO Status

When GPIO1 is not used as a LCD signal and GPIO1 is configured as an output, writing a 1 to this bit drives GPIO1 high and writing a 0 to this bit drives GPIO1 low.
When GPIO1 is not used as a LCD signal and GPIO1 is configured as an input, a read from this bit returns the status of GPIO1.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 1 is written to this bit, the D-TFD signal YSCL signal is enabled.
When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) and a 0 is written to this bit, the D-TFD signal YSCL signal is forced low.

When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) and a 1 is written to this bit, the HR-TFT signal CLS signal is enabled.
When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) and a 0 is written to this bit, the HR-TFT signal CLS signal is forced low.

When a TFT Type 2 panel is enabled (REG[C5h] bits 1:0 = 01) and a 1 is written to this bit, the AP signal is enabled.
When a TFT Type 2 panel is enabled (REG[C5h] bits 1:0 = 01) and a 0 is written to this bit, the AP signal is forced low.

When a TFT Type 3 panel is enabled (REG[C5h] bits 1:0 = 10) and a 1 is written to this bit, the OE signal is enabled.
When a TFT Type 3 panel is enabled (REG[C5h] bits 1:0 = 10) and a 0 is written to this bit, the OE signal is forced low.

bit 0 GPIO0 Pin IO Status

When GPIO1 is not used as a LCD signal and GPIO0 is configured as an output, writing a 1 to this bit drives GPIO0 high and writing a 0 to this bit drives GPIO0 low.
When GPIO1 is not used as a LCD signal and GPIO0 is configured as an input, a read from this bit returns the status of GPIO0.

When a D-TFD panel is enabled (REG[10h] bits 1:0 = 11) writing to this bit has no effect.

When a HR-TFT panel is enabled (REG[10h] bits 1:0 = 10) writing to this bit has no effect.

When a TFT Type 2 panel is enabled (REG[C5h] bits 1:0 = 01) writing to this bit has no effect.

When a TFT Type 3 panel is enabled (REG[C5h] bits 1:0 = 10) writing to this bit has no effect.

General Purpose IO Pins Status/Control Register 1
REG[ADh] Read/Write

7	6	5	4	3	2	1	0
GPO0 Control	n/a						

bit 7 GPO0 Control
This bit controls the General Purpose Output 0 pin.
Writing a 0 to this bit drives GPO0 to low.
Writing a 1 to this bit drives GPO0 to high.

This bit has no effect when HR-TFT panel type is selected.

Note
Many implementations use the GPO0 pin to control the LCD bias power (see Section 6.3, “LCD Power Sequencing” on page 66).

8.3.9 Pulse Width Modulation (PWM) Clock and Contrast Voltage (CV) Pulse Configuration Registers

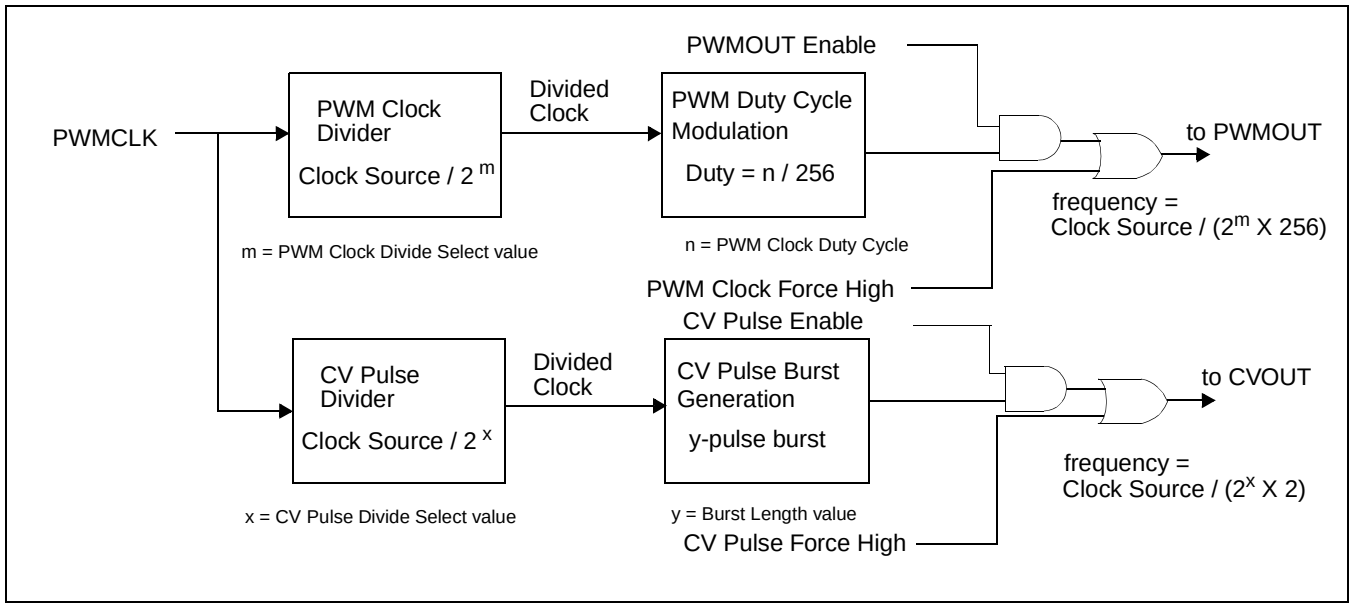


Figure 8-2 PWM Clock/CV Pulse Block Diagram

Note
For further information on PWMCLK, see Section 7.1.4, “PWMCLK” on page 115.

PWM Clock / CV Pulse Control Register
REG[B0h] Read/Write

PWM Clock Force High	n/a		PWM Clock Enable	CV Pulse Force High	CV Pulse Burst Status (RO)	CV Pulse Burst Start	CV Pulse Enable
7	6	5	4	3	2	1	0

bit 7 and bit 4 PWM Clock Force High (bit 7) and PWM Clock Enable (bit 4)
These bits control the PWMOUT pin and PWM Clock circuitry as follows.

Table 8-15: PWM Clock Control

Bit 7	Bit 4	Result
0	1	PWM Clock circuitry enabled (controlled by REG[B1h] and REG[B3h])
0	0	PWMOUT forced low
1	x	PWMOUT forced high

x = don’t care

When PWMOUT is forced low or forced high it can be used as a general purpose output.

Note
The PWM Clock circuitry is disabled when Power Save Mode is enabled.

bit 3 and bit 0 CV Pulse Force High (bit 3) and CV Pulse Enable (bit 0)
These bits control the CVOUT pin and CV Pulse circuitry as follows.

Table 8-16: CV Pulse Control

Bit 3	Bit 0	Result
0	1	CV Pulse circuitry enabled (controlled by REG[B1h] and REG[B2h])
0	0	CVOUT forced low
1	x	CVOUT forced high

x = don't care

When CVOUT is forced low or forced high it can be used as a general purpose output.

Note
Bit 0 must be set to 1 before initiating a new burst using the CV Pulse Burst Start bit.

Note
The CV Pulse circuitry is disabled when Power Save Mode is enabled.

bit 2 CV Pulse Burst Status
This is a read-only bit. A "1" indicates a CV pulse burst is occurring. A "0" indicates no CV pulse burst is occurring. Software should wait for this bit to clear before starting another burst.

bit 1 CV Pulse Burst Start
A 1 in this bit initiates a single CVOUT pulse burst. The number of clock pulses generated is programmable from 1 to 256. The frequency of the pulses is the divided CV Pulse source divided by 2, with 50/50 duty cycle. This bit should be cleared to 0 by software before initiating a new burst.

Note
This bit has effect only if the CV Pulse Enable bit is 1.

bit 0 CV Pulse Enable
See description for bit 3.

PWM Clock / CV Pulse Configuration Register
REG[B1h] Read/Write

PWM Clock Divide Select Bits 3-0				CV Pulse Divide Select Bits 2-0			PWMCLK Source Select
7	6	5	4	3	2	1	0

bits 7-4 PWM Clock Divide Select Bits [3:0]
The value of these bits represents the power of 2 by which the selected PWM clock source is divided.

Table 8-17: PWM Clock Divide Select Options

PWM Clock Divide Select Bits [3:0]	PWM Clock Divide Amount
0h	1
1h	2
2h	4
3h	8
...	...
Ch	4096
Dh-Fh	Reserved

Note
This divided clock is further divided by 256 before it is output at PWMOUT.

bits 3-1 CV Pulse Divide Select Bits [2:0]
The value of these bits represents the power of 2 by which the selected CV Pulse source is divided.

Table 8-18: CV Pulse Divide Select Options

CV Pulse Divide Select Bits [2:0]	CV Pulse Divide Amount
0h	1
1h	2
2h	4
3h	8
...	...
7h	128

Note
This divided clock is further divided by 2 before it is output at the CVOUT.

bit 0 PWMCLK Source Select
When this bit = 0, the clock source for PWMCLK is the BCLK source.
When this bit = 1, the clock source for PWMCLK is CLKI2 or XTAL (see BCLK Source Select Register on page 157).

Note
For further information on the PWMCLK source select, see Section 7.2, “Clock Selection” on page 116.

CV Pulse Burst Length Register
REG[B2h] Read/Write

CV Pulse Burst Length Bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 CV Pulse Burst Length Bits [7:0]
The value of this register determines the number of pulses generated in a single CV Pulse burst:

Number of pulses in a burst = (ContentsOfThisRegister) + 1

PWMOUT Duty Cycle Register
REG[B3h] Read/Write

PWMOUT Duty Cycle Bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 PWMOUT Duty Cycle Bits [7:0]
This register determines the duty cycle of the PWMOUT output.

Table 8-19: PWMOUT Duty Cycle Select Options

PWMOUT Duty Cycle [7:0]	PWMOUT Duty Cycle
00h	Always Low
01h	High for 1 out of 256 clock periods
02h	High for 2 out of 256 clock periods
...	...
FFh	High for 255 out of 256 clock periods

8.3.10 Extended Registers

Memory Access Pointer 0
REG[C0h] Read/Write

7	6	5	4	3	2	1	0
Memory Access Pointer Bits 7-0							

Memory Access Pointer 1
REG[C1h] Read/Write

7	6	5	4	3	2	1	0
Memory Access Pointer Bits 15-8							

Memory Access Pointer 2
REG[C2h] Read/Write

7	6	5	4	3	2	1	0
n/a							Memory Access Pointer Bit 16

bits 16-0 Memory Access Pointer Bits [16:0]

These registers control memory accesses for the Indirect Interface only (CNF[2:0] = 111). These bits contain a pointer to the address position in the display buffer (memory) used when a "data" read/write is executed. At reset, these registers are set to 0. After each byte read/write, the Memory Access Pointer is incremented by 1. After each word read/write, the Memory Access Pointer is incremented by 2. For further information on accessing the S1D13708 display buffer, see Section 17, "Big-Endian Bus Interface" on page 212.

Note

[1] If the Memory Access Pointer programmed specifies an odd memory address, only byte accesses may be performed.

[2] These bits take effect only after a "command" write to the Memory Access Start register (REG[C4h]).

Memory Access Start
REG[C4h] Write only

7	6	5	4	3	2	1	0
n/a							

bits 7-0 Memory Access Start [7:0]

This register controls memory accesses for the Indirect Interface only (CNF[2:0] = 111). This register is the trigger which readies the interface for reads/writes from/to the display buffer. After a "command" write to this register, successive reads/writes may be performed. After each byte read/write, the Memory Access Pointer is incremented by 1. After each word read/write, the Memory Access Pointer is incremented by 2.

Extended Panel Type Register
REG[C5h] Read/Write

n/a						Extended Panel Type Bits 1-0	
7	6	5	4	3	2	1	0

bits 1-0 Extended Panel Type Bits [1:0]
These bits override the setting in REG[10h] bits 1-0 and allow selection of the alternate TFT panel types.

Table 8-20: Extended Panel Type Selection

REG[C5h] Bits [1:0]	Panel Type
00	no effect from REG[10h] bits 1-0
01	TFT Type 2
10	TFT Type 3
11	TFT Type 4

Memory Access Select Register
REG[C6h] Read/Write

n/a							Memory Access Select
7	6	5	4	3	2	1	0

bit 0 Memory Access Select Bit
This register only has an effect when the Indirect Interface is enabled (CNF[2:0] = 111). This bit selects the type of memory access, byte or word, used for memory accesses when the Indirect Interface is enabled.
When this bit = 0, byte or word accesses are allowed. For Mode 68, EBL and EBU indicate the transfer size. For Mode 80, WRU#, WRL#, RDU#, and RDL# indicate the transfer size.
When this bit = 1, only word accesses are allowed and WRU#, RDU#, EBU are ignored when determining the transfer size. This bit affects memory accesses only, the signals are not ignored for register accesses.

Note
All word accesses must be byte aligned on even addresses.

Ink Layer Transparent Color Register 0
REG[C7h] Read/Write

Ink Layer Transparent Color Bits 7-0							
7	6	5	4	3	2	1	0

Ink Layer Transparent Color Register 1
REG[C8h] Read/Write

Ink Layer Transparent Color Bits 15-8							
7	6	5	4	3	2	1	0

bits 15-0 Ink Layer Transparent Color Bits [15:0]
The Ink Layer requires a transparent color to be selected. This transparent color is compared with the values in the foreground image during display refresh. If the pixel value matches the transparent color, the pixel from the background image is shown. If the pixel value does not match the transparent color, the foreground image is shown.

For further information on the Ink Layer, see Section 14, “Ink Layer” on page 187.

Ink Layer Register
REG[C9h] Read/Write

7	6	5	4	3	2	1	0
n/a							Ink Layer Enable

bit 0 Ink Layer Enable
This bit controls whether the Ink Layer is enabled or disabled.
When this bit = 0, the Ink Layer is disabled.
When this bit = 1, the Ink Layer is enabled.

For further information on the Ink Layer, see Section 14, “Ink Layer” on page 187.

Note
Picture-in-Picture Plus (PIP$^+$) is not available when the Ink Layer is enabled.

BCLK Source Select Register
REG[CAh] Read/Write

7	6	5	4	3	2	1	0
n/a		XTAL Enable Bits 1-0		n/a		Extended PCLK/PWMCLK Source Select	BCLK Source Select

bits 5-4 XTAL Enable Bits [1:0]
These two bits enable the crystal input/drive circuitry.
When these bits = 00, the XTAL circuitry is disabled.
When these bits = 11, the XTAL circuitry is enabled.

For further information on the crystal usage, see Section 16, “Embedded Crystal Oscillator” on page 211.

bit 1 Extended PCLK / PWMCLK Source Select
This bit selects the source (between CLKI2 and XTAL) used for PCLK and the PWM clock when CLKI2 is selected as the source for PCLK (see REG[05h] bits 1-0) and PWM clock (see REG[B1h] bit 0).
When this bit = 0, the source used for PCLK and PWMCLK is CLKI2.
When this bit = 1, the source used for PCLK and PWMCLK is XTAL.

bit 0 BCLK Source Select

This bit selects the BCLK source between CLKI and XTAL (XTAL is recommended only when configured for the Indirect Interface, see CNF[2:0]). When this bit is used the BCLK source switches in a glitch free manner.
When this bit = 0, the BCLK source is CLKI.
When this bit = 1, the BCLK source is XTAL.

Note
There must be a CLKI source initially for this bit to operate.

Note
The BCLK source may be divided down using the BCLK Source to BCLK divide ratio select (CNF[7:6]).

Data Compare Invert Enable Register
REG[CBh] Read/Write

7	6	5	4	3	2	1	0
n/a							Data Compare Invert Enable

bit 0 Data Compare Invert Enable

This bit can be used to lower power consumption for TFT Type 2 and TFT Type 3 Interfaces. The Data Compare and Invert function reduces the amount of data toggled by counting the number of bits that are changed (1 to 0 **or** 0 to 1) from the previous pixel data. For all other panel interfaces it has no effect.
When this bit = 0, the Data Compare and Invert functions are disabled.
When this bit = 1, the Data Compare and Invert functions are enabled.

TFT Type 2 VCLK Configuration Register
REG[D0h] Read/Write

n/a			VCLK Hold bits 1-0		n/a	VCLK Setup bits 1-0	
7	6	5	4	3	2	1	0

bits 4-3 VCLK Hold Bits [1:0]
These bits control the TFT Type 2 AC timing parameter from the rising edge of STB to the falling edge of VCLK. The parameter is selected as follows. For all other panel interfaces it has no effect.

Table 8-21: VCLK Hold

REG[D0h] bits 1-0	VCLK Hold (in PCLKs)
00	**7**
01	**9**
10	**12**
11	**16**

bits 1-0 VCLK Setup Bits [1:0]
These bits control the TFT Type 2 AC timing parameter from the rising edge of VCLK to the rising edge of STB. The parameter is selected as follows. For all other panel interfaces it has no effect.

Table 8-22: VCLK Setup

REG[D0h] bits 4-3	VCLK Setup (in PCLKs)
00	**7**
01	**9**
10	**12**
11	**16**

TFT Type 2 AP Configuration Register
REG[D1h] Read/Write

POL Type	n/a	AP Pulse Width bits 2-0			n/a	AP Rising Position bits 1-0	
7	6	5	4	3	2	1	0

bit 7 POL Type
This bit selects how often the POL signal is toggled. The POL signal is used for the TFT Type 2 Interface. For all other panel interfaces this bit has no effect.
When this bit = 0, the POL signal is toggled every line.
When this bit = 1, the POL signal is toggled every frame.

bits 5-3 AP Pulse Width Bits [2:0]
These bits specify the AP Pulse Width used for the TFT Type Interface. For all other panel interfaces it has no effect.

Table 8-23: AP Pulse Width

REG[D1h] bits 5-3	AP Pulse Width (in PCLKs)
000	20
001	40
010	80
011	120
100	150
101	190
110	240
111	270

bits 1-0 AP Rising Position Bits [1:0]
These bits control the TFT Type 2 AC timing parameter from the rising edge of STB to the rising edge of AP. The parameter is selected as follows. For all other panel interfaces it has no effect.

Table 8-24: AP Rising Position

REG[D1h] bits 1-0	GPIO1 (AP) Rising Position (in PCLKs)
00	40
01	52
10	68
11	90

TFT Type 3 Signal Control Register
REG[D4h] Read/Write

PDME Control	XSTBY Control	XOHV Control	XRESV Control	XRESH Control	PCLK2 Enable	PCLK1 Enable	n/a
7	6	5	4	3	2	1	0

Note
The GPO pins are used by the TFT Type 3 interface when REG[C5h] bits 1-0 = 10. For pin mapping for TFT Type 3, see Table 4-10: "LCD Interface Pin Mapping," on page 40.

bit 7 PDME Control
If the TFT Type 3 interface is selected (REG[C5h] bits 1-0 = 10), this bit controls the LCD signal PDME.
When this bit = 1, PDME = 1.
When this bit = 0, PDME = 0.

bit 6 XSTBY Control
If the TFT Type 3 interface is selected (REG[C5h] bits 1-0 = 10), this bit controls the LCD signal XSTBY.
When this bit = 1, XSTBY = 1.
When this bit = 0, XSTBY = 0.

bit 5 XOHV Control
If the TFT Type 3 interface is selected (REG[C5h] bits 1-0 = 10), this bit controls the LCD signal XOHV.
When this bit = 1, XOHV = 1.
When this bit = 0, XOHV = 0.

bit 4 XRESV Control
If the TFT Type 3 interface is selected (REG[C5h] bits 1-0 = 10), this bit controls the LCD signal XRESV.
When this bit = 1, XRESV = 1.
When this bit = 0, XRESV = 0.

bit 3 XRESH Control
If the TFT Type 3 interface is selected (REG[C5h] bits 1-0 = 10), this bit controls the LCD signal XRESH.
When this bit = 1, XRESH = 1.
When this bit = 0, XRESH = 0.

bit 2 PCLK2 Control
If the TFT Type 3 interface is selected (REG[C5h] bits 1-0 = 10), this bit enables the LCD signal PCLK2.
When this bit = 1, PCLK2 = 1.
When this bit = 0, PCLK2 = 0.

bit 1 PCLK1 Control

If the TFT Type 3 interface is selected (REG[C5h] bits 1-0 = 10), this bit enables the LCD signal PCLK1.

When this bit = 1, PCLK1 = 1.

When this bit = 0, PCLK1 = 0.

TFT Type 3 OE Rising Edge Position Register

REG[D5h] Read/Write

OE Rising Edge Position bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 OE Rising Edge Position Bits [7:0]

These bits specify the rising edge position of the OE signal in 2 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 OE Pulse Width Register

REG[D6h] Read/Write

OE Pulse Width bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 OE Pulse Width Bits [7:0]

These bits specify the pulse width of the OE signal in 2 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 POL Toggle Position Register

REG[D7h] Read/Write

POL Toggle Position bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 POL Toggle Position Bits [7:0]

These bits specify the toggle position of the POL signal in 2 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 VCOM Toggle Position Register

REG[D8h] Read/Write

VCOM Toggle Position bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 VCOM Toggle Position Bits [7:0]

These bits specify the toggle position of the VCOM signal in 2 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 CPV Pulse Width Register
REG[D9h] Read/Write

CPV Pulse Width bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 CPV Pulse Width Bits [7:0]
These bits specify the pulse width of the CPV signal in 2 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 XOEV Rising Edge Position Register
REG[DAh] Read/Write

XOEV Rising Edge Position bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 XOEV Rising Edge Position Bits [7:0]
These bits specify the rising edge position of the XOEV signal in 2 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces. The value of this register must be greater than zero.

TFT Type 3 XOEV Falling Edge Position Register
REG[DBh] Read/Write

XOEV Falling Edge Position bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 XOEV Falling Edge Position Bits [7:0]
These bits specify the falling edge position of the XOEV signal in 2 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 PCLK Divide Register
REG[DCh] Read/Write

n/a		PCLK2 Divide Rate		PCLK1 Divide Rate			
7	6	5	4	3	2	1	0

bit 5-4 PCLK2 Divide Rate Bits [1:0]
These bits specify the divide rate for PCLK2. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

Table 8-25: GPO2 PCLK2 Divide Rate

REG[DBh] bits 5-4	PCLK2 Divide Rate
00	64
01	128
10	256
11	512

bits 3-0 PCLK1 Divide Rate Bits [3:0]
These bits specify the divide rate for PCLK1. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

Table 8-26: GPO1 PCLK1 Divide Rate

REG[DBh] bits 3-0	GPO1 PCLK1 Divide Rate
0000	2
0001	4
0010	8
0011	16
0100	32
0101	64
0110	128
0111	256
1000	512
1001	1024
1010	2048
1011	4096
1100	8192
1101	16384
1110	32768
1111	65536

TFT Type 3 Partial Mode Display Area Control Register
REG[E0h] Read/Write

n/a			Partial Mode Display Enable	Partial Mode Display Type Select	Area 2 Display Enable	Area 1 Display Enable	Area 0 Display Enable
7	6	5	4	3	2	1	0

bit 4 Partial Mode Display Enable
This bit enables/disables the Partial Mode Display for the TFT Type 3 and has no effect for all other panel interfaces.
When this bit = 1, Partial Mode Display is enabled.
When this bit = 0, Partial Mode Display is disabled.

bit 3 Partial Mode Display Type Select
This bit selects the type of partial mode display.
When this bit =0, the Stripe type of partial mode display is selected. If Stripe is enabled only the Y Position registers are used in calculating the partial display.
When this bit = 1, type Block type of partial mode display is selected. If Block is enabled both the X and Y Position registers are used in calculating the partial display.

bit 2 Area 2 Display Enable
This bit enables/disables the Area 2 for Partial Mode Display on the TFT Type 3 and has no effect for all other panel interfaces.
When this bit = 1, Area 2 is enabled.
When this bit = 0, Area 2 is disabled.

bit 1 Area 1 Display Enable
This bit enables/disables the Area 1 for Partial Mode Display on the TFT Type 3 and has no effect for all other panel interfaces.
When this bit = 1, Area 1 is enabled.
When this bit = 0, Area 1 is disabled.

bit 0 Area 0 Display Enable
This bit enables/disables the Area 0 for Partial Mode Display on the TFT Type 3 and has no effect for all other panel interfaces.
When this bit = 1, Area 0 is enabled.
When this bit = 0, Area 0 is disabled.

TFT Type 3 Partial Mode Display Refresh Cycle Register
REG[E1h] Read/Write

n/a		Partial Mode Display Refresh Cycle bits 5-0					
7	6	5	4	3	2	1	0

bits 5-0 Partial Mode Display Refresh Cycle Bits [5:0]
These bits specify the refresh cycle for the Partial Mode Display. The refresh cycle can be a value from 0 to 63. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 Partial Area 0 X Start Position Register
REG[E2h] Read/Write

7	6	5	4	3	2	1	0
n/a		Partial Area 0 X Start Position bits 5-0					

bits 5-0 Partial Area 0 X Start Position Bits [5:0]
These bits specify the X Start Position of Partial Area 0 in 8 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 Partial Area 0 Y Start Position Register
REG[E3h] Read/Write

7	6	5	4	3	2	1	0
n/a		Partial Area 0 Y Start Position bits 5-0					

bits 5-0 Partial Area 0 Y Start Position Bits [5:0]
These bits specify the Y Start Position of Partial Area 0 in 8 line resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 Partial Area 0 X End Position Register
REG[E4h] Read/Write

7	6	5	4	3	2	1	0
n/a		Partial Area 0 X End Position bits 5-0					

bits 5-0 Partial Area 0 X End Position Bits [5:0]
These bits specify the X End Position of Partial Area 0 in 8 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 Partial Area 0 Y End Position Register
REG[E5h] Read/Write

7	6	5	4	3	2	1	0
n/a		Partial Area 0 Y End Position bits 5-0					

bits 5-0 Partial Area 0 Y End Position Bits [5:0]
These bits specify the Y End Position of Partial Area 0 in 8 line resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 Partial Area 1 X Start Position Register
REG[E6h] Read/Write

n/a		Partial Area 1 X Start Position bits 5-0					
7	6	5	4	3	2	1	0

bits 5-0 Partial Area 1 X Start Position Bits [5:0]
These bits specify the X Start Position of Partial Area 1 in 8 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 Partial Area 1 Y Start Position Register
REG[E7h] Read/Write

n/a		Partial Area 1 Y Start Position bits 5-0					
7	6	5	4	3	2	1	0

bits 5-0 Partial Area 1 Y Start Position Bits [5:0]
These bits specify the Y Start Position of Partial Area 1 in 8 line resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 Partial Area 1 X End Position Register
REG[E8h] Read/Write

n/a		Partial Area 1 X End Position bits 5-0					
7	6	5	4	3	2	1	0

bits 5-0 Partial Area 1 X End Position Bits [5:0]
These bits specify the X End Position of Partial Area 1 in 8 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 Partial Area 1 Y End Position Register
REG[E9h] Read/Write

n/a		Partial Area 1 Y End Position bits 5-0					
7	6	5	4	3	2	1	0

bits 5-0 Partial Area 1 Y End Position Bits [5:0]
These bits specify the Y End Position of Partial Area 1 in 8 line resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 Partial Area 2 X Start Position Register
REG[EAh] Read/Write

7	6	5	4	3	2	1	0
n/a		Partial Area 2 X Start Position bits 5-0					

bits 5-0 Partial Area 2 X Start Position Bits [5:0]
These bits specify the X Start Position of Partial Area 2 in 8 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 Partial Area 2 Y Start Position Register
REG[EBh] Read/Write

7	6	5	4	3	2	1	0
n/a		Partial Area 2 Y Start Position bits 5-0					

bits 5-0 Partial Area 2 Y Start Position Bits [5:0]
These bits specify the Y Start Position of Partial Area 2 in 8 line resolution. This register is used for the TFT Type 3Interface and has no effect for all other panel interfaces.

TFT Type 3 Partial Area 2 X End Position Register
REG[ECh] Read/Write

7	6	5	4	3	2	1	0
n/a		Partial Area 2 X End Position bits 5-0					

bits 5-0 Partial Area 2 X End Position Bits [5:0]
These bits specify the X End Position of Partial Area 2 in 8 pixel resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 Partial Area 2 Y End Position Register
REG[EDh] Read/Write

7	6	5	4	3	2	1	0
n/a		Partial Area 2 Y End Position bits 5-0					

bits 5-0 Partial Area 2 Y End Position Bits [5:0]
These bits specify the Y End Position of Partial Area 2 in 8 line resolution. This register is used for the TFT Type 3 Interface and has no effect for all other panel interfaces.

TFT Type 3 Command 0 Store Register 0
REG[F0h] Read/Write

Command 0 Store bits 7-0							
7	6	5	4	3	2	1	0

TFT Type 3 Command 0 Store Register 1
REG[F1h] Read/Write

n/a				Command 0 Store bits 11-8			
7	6	5	4	3	2	1	0

bits 11-0 Command 0 Store Bits [11:0]
These bits store command 0 for the TFT Type 3 Interface. This register has no effect for all other panel interfaces.

TFT Type 3 Command 1 Store Register 0
REG[F2h] Read/Write

Command 1 Store bits 7-0							
7	6	5	4	3	2	1	0

TFT Type 3 Command 1 Store Register 1
REG[F3h] Read/Write

n/a				Command 1 Store bits 11-8			
7	6	5	4	3	2	1	0

bits 11-0 Command 1 Store Bits [11:0]
These bits store command 1 for the TFT Type 3 Interface. This register has no effect for all other panel interfaces.

TFT Type 3 Command Send Request Register
REG[F4h] Read/Write

n/a							Command Send Request
7	6	5	4	3	2	1	0

bit 0 Command Send Request
After the CPU sets this bit, the S1D13708 sends the command in the next non-display period and clears this bit automatically. This register has no effect for all other panel interfaces.

TFT Type 3 Source Driver IC Number Register
REG[F5h] Read/Write

n/a						Source Driver IC Number Bits 1-0	
7	6	5	4	3	2	1	0

bits 1-0 Source Driver IC Number Bits [1:0]
These bits contain the number of Source Driver ICs.

Table 8-27: Number of Source Driver ICs

REG[F5h] bits 1-0	Source Driver ICs
00	**1**
01	**2**
10	**3**
11	**4**

9 Frame Rate Calculation

The following formula is used to calculate the display frame rate.

$$\text{FrameRate} = \frac{f_{PCLK}}{(HT) \times (VT)}$$

Where:

f_{PCLK} = PClk frequency (Hz)

HT = Horizontal Total
= ((REG[12h] bits 6-0) + 1) x 8 Pixels

VT = Vertical Total
= ((REG[19h] bits 1-0, REG[18h] bits 7-0) + 1) Lines

10 Display Data Formats

The following diagrams show the display mode data formats for a little-endian system.

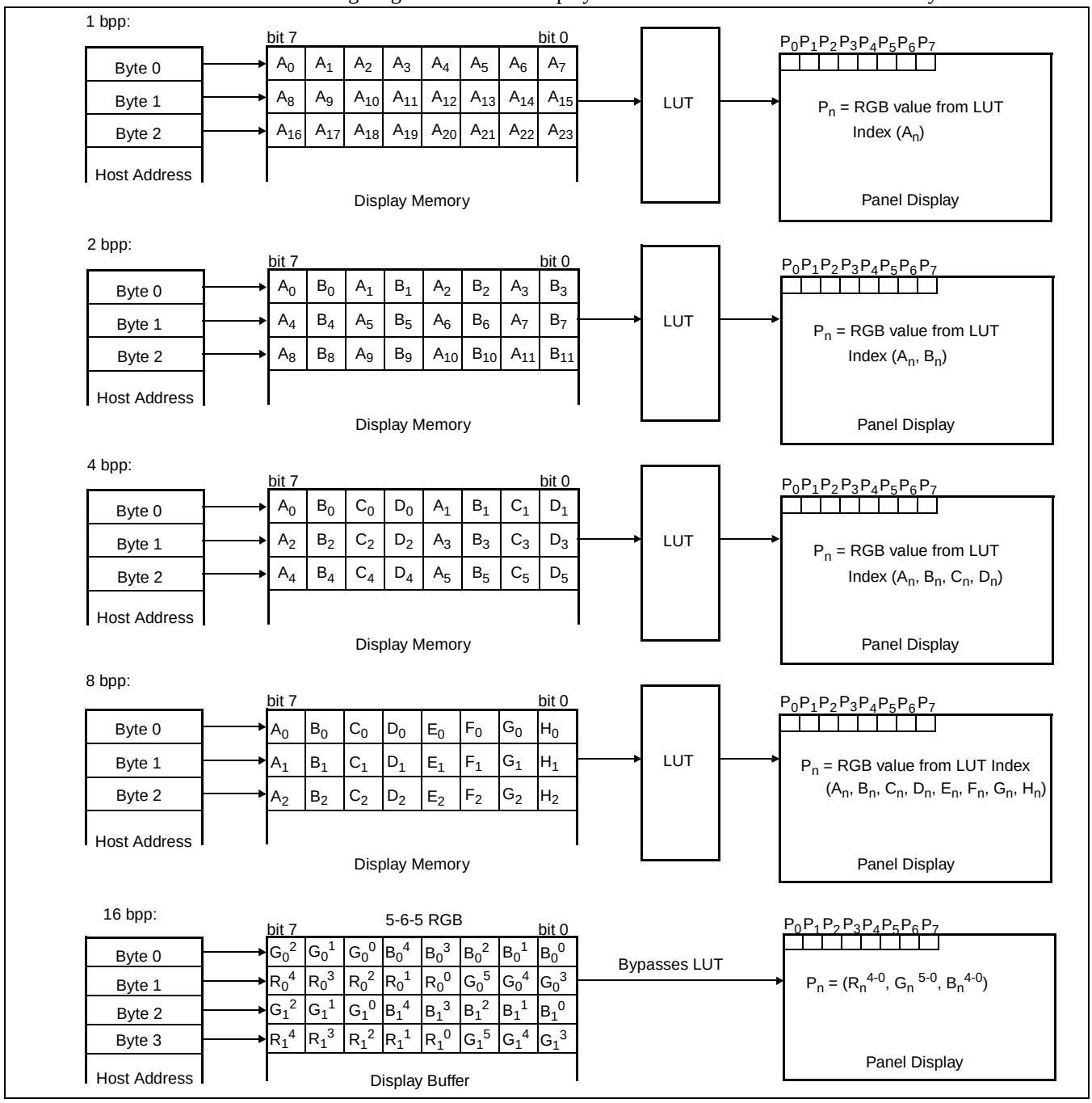


Figure 10-1 4/8/16 Bit-Per-Pixel Display Data Memory Organization

Note

1. The Host-to-Display mapping shown here is for a little endian system.
2. For 16 bpp format, R_n, G_n, B_n represent the red, green, and blue color components.

11 Look-Up Table Architecture

The following figures are intended to show the display data output path only.

Note
When Video Data Invert is enabled the video data is inverted after the Look-Up Table.

11.1 Monochrome Modes

The green Look-Up Table (LUT) is used for all monochrome modes.

1 Bit-per-pixel Monochrome Mode

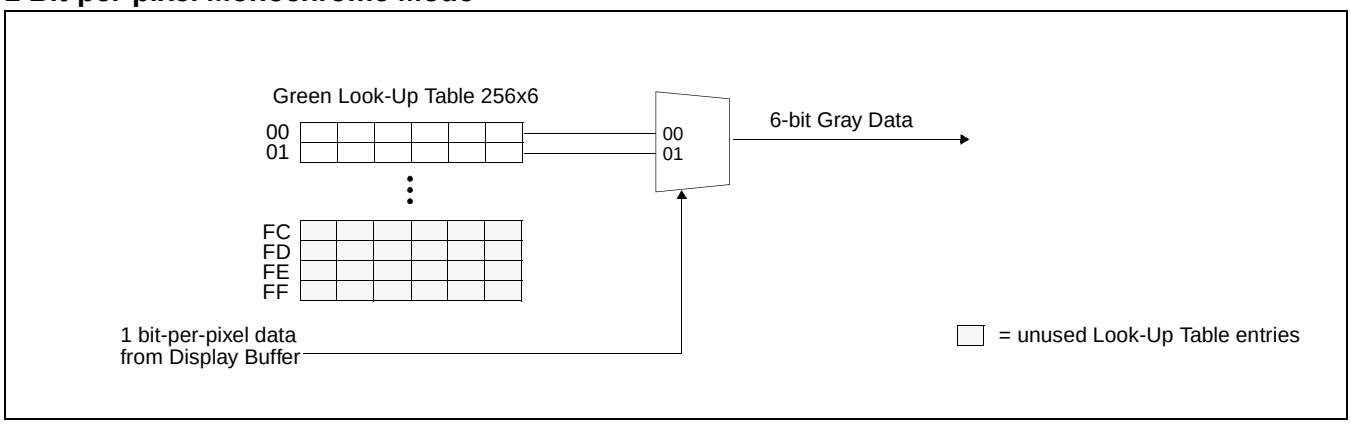


Figure 11-1 1 Bit-per-pixel Monochrome Mode Data Output Path

2 Bit-per-pixel Monochrome Mode

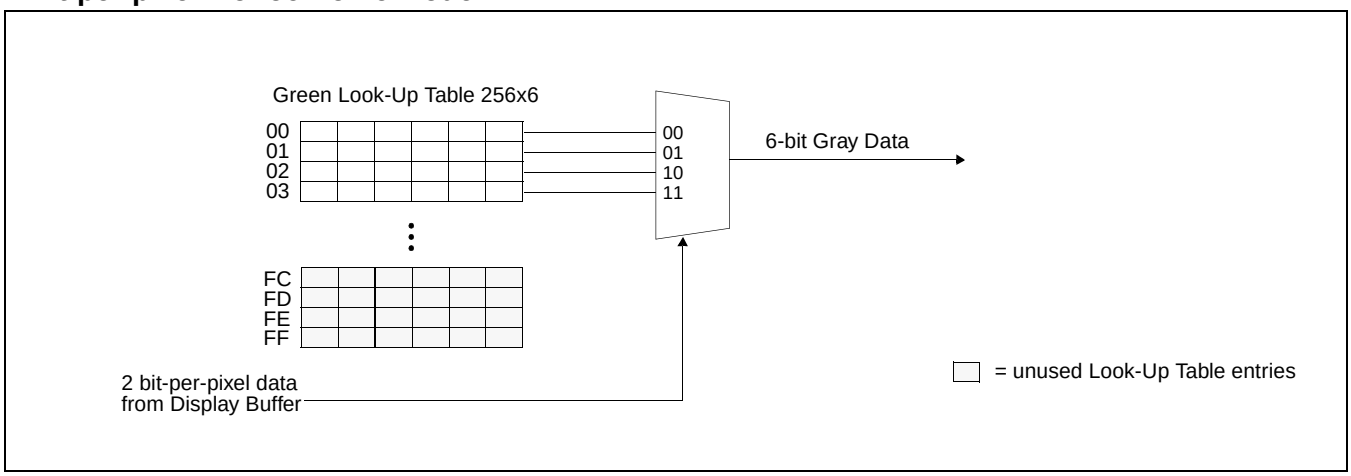


Figure 11-2 2 Bit-per-pixel Monochrome Mode Data Output Path

4 Bit-per-pixel Monochrome Mode

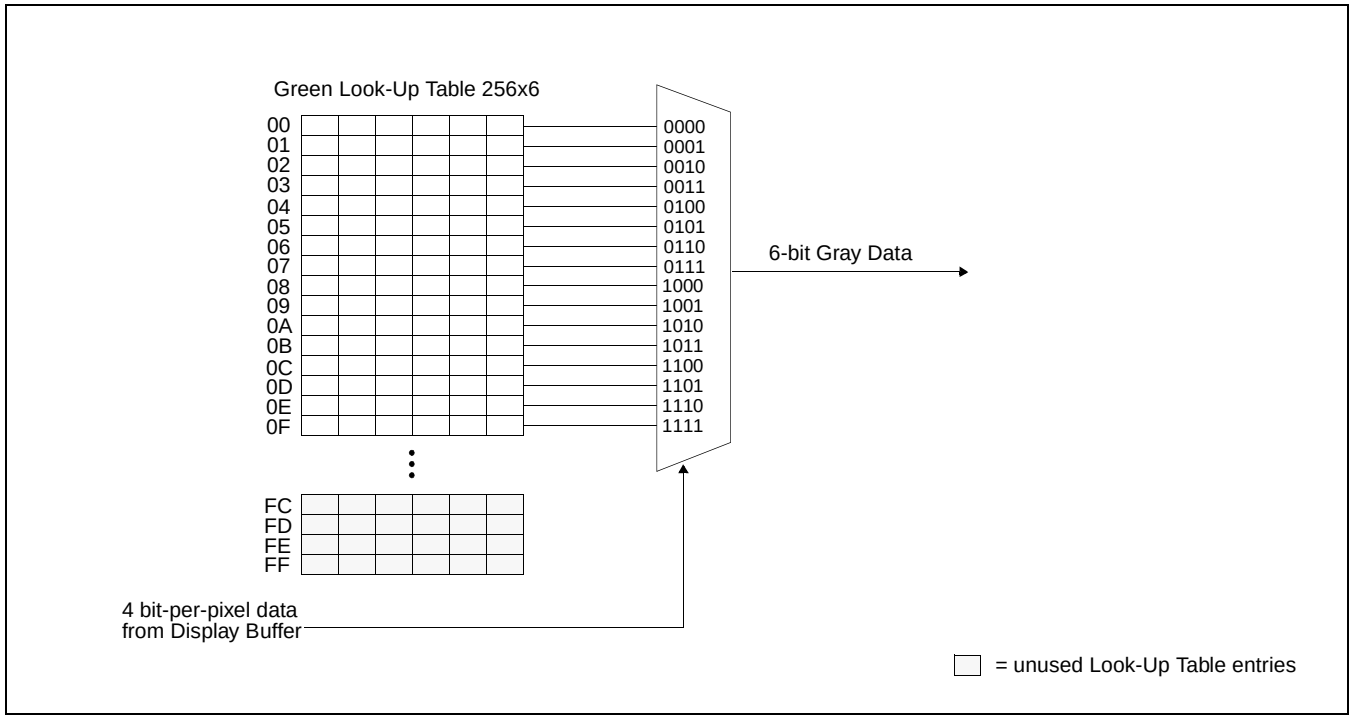


Figure 11-3 4 Bit-per-pixel Monochrome Mode Data Output Path

8 Bit-per-pixel Monochrome Mode

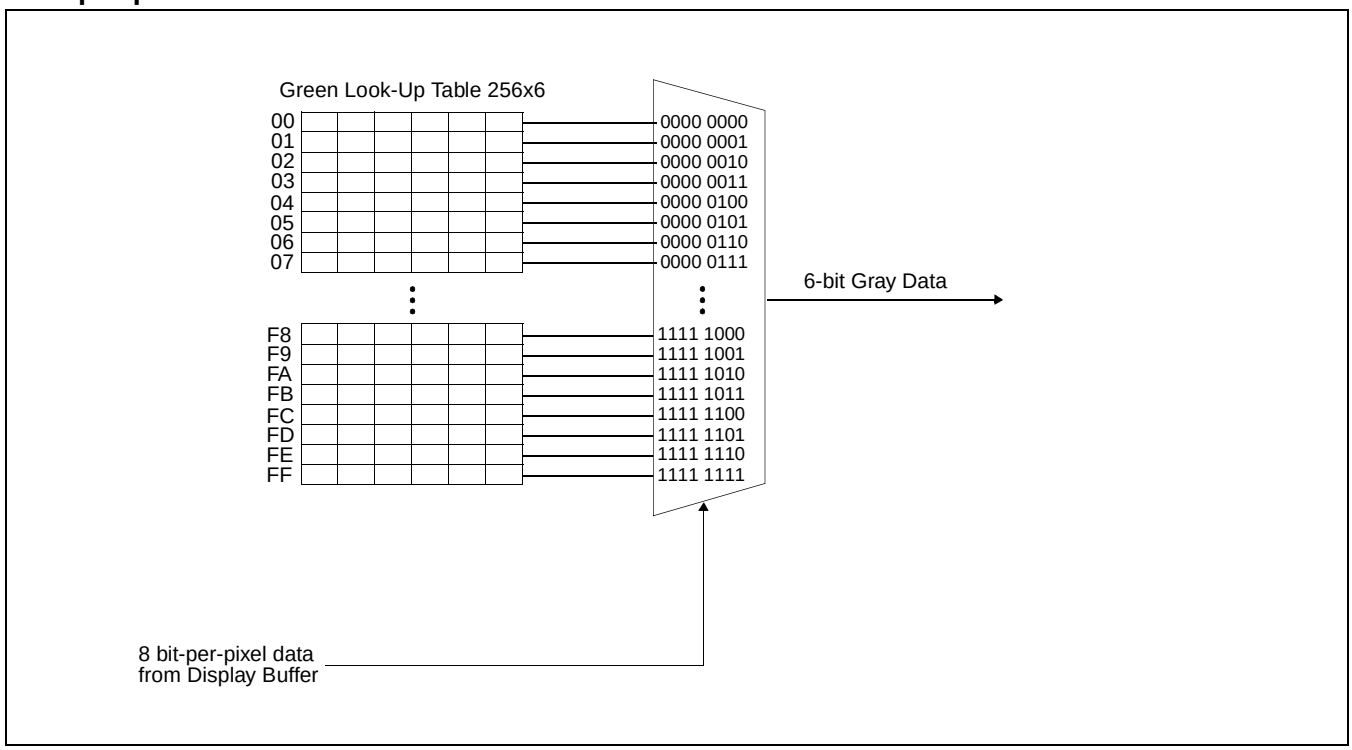


Figure 11-4 8 Bit-per-pixel Monochrome Mode Data Output Path

16 Bit-Per-Pixel Monochrome Mode

The LUT is bypassed and the green data is directly mapped for this color depth– See “Display Data Formats” on page 172..

11.2 Color Modes

1 Bit-Per-Pixel Color

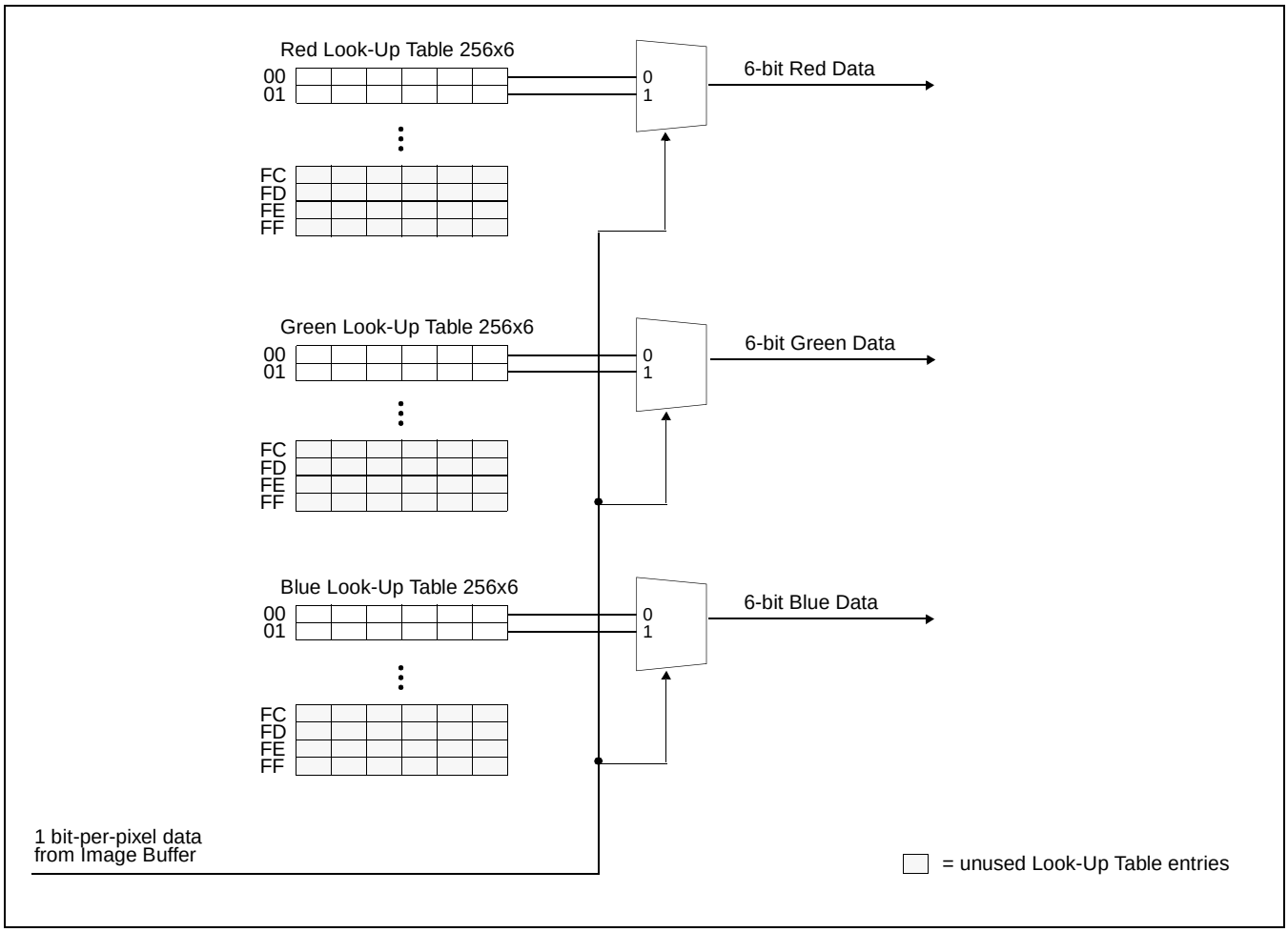


Figure 11-5 1 Bit-Per-Pixel Color Mode Data Output Path

2 Bit-Per-Pixel Color

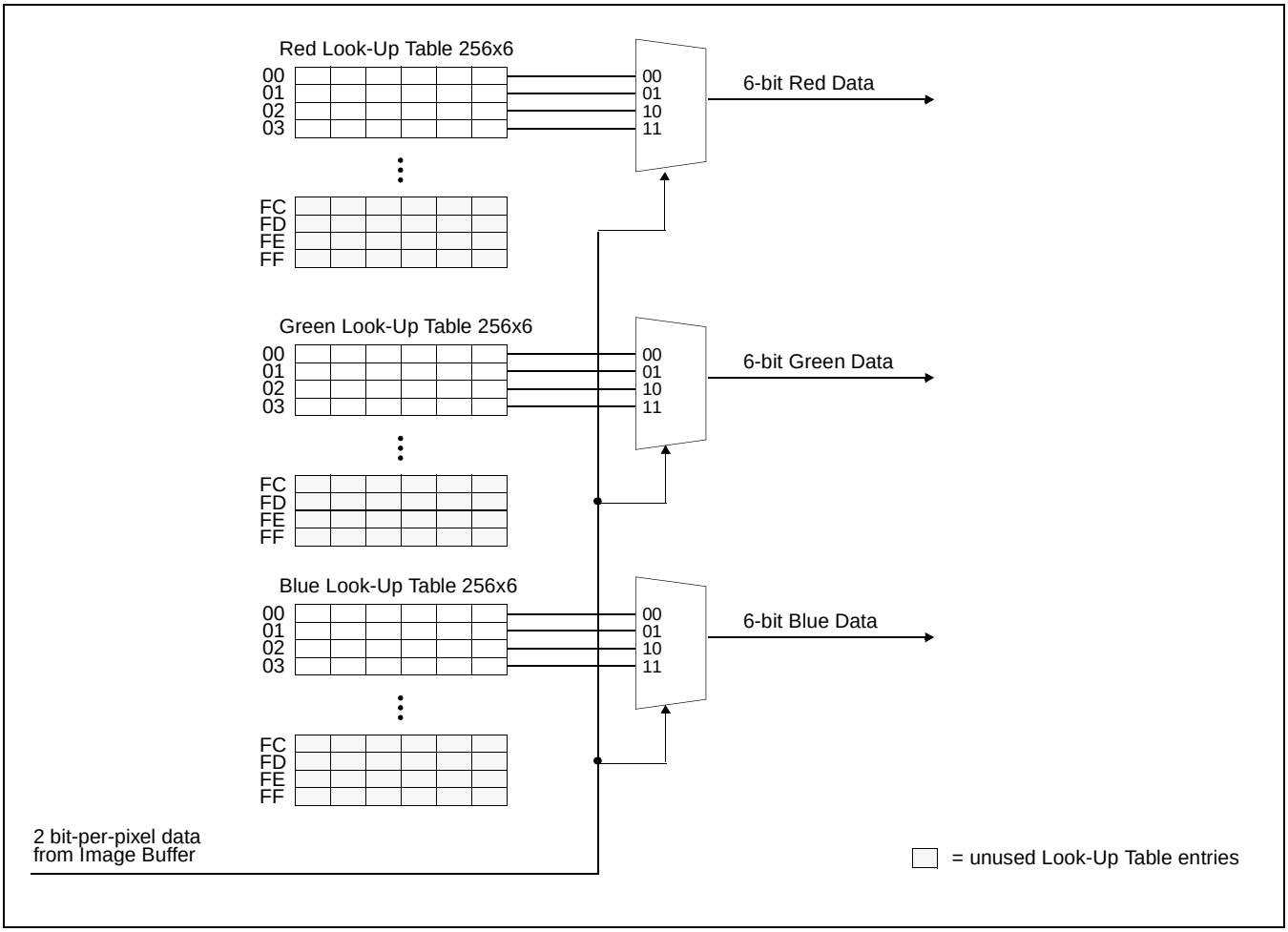


Figure 11-6 2 Bit-Per-Pixel Color Mode Data Output Path

4 Bit-Per-Pixel Color

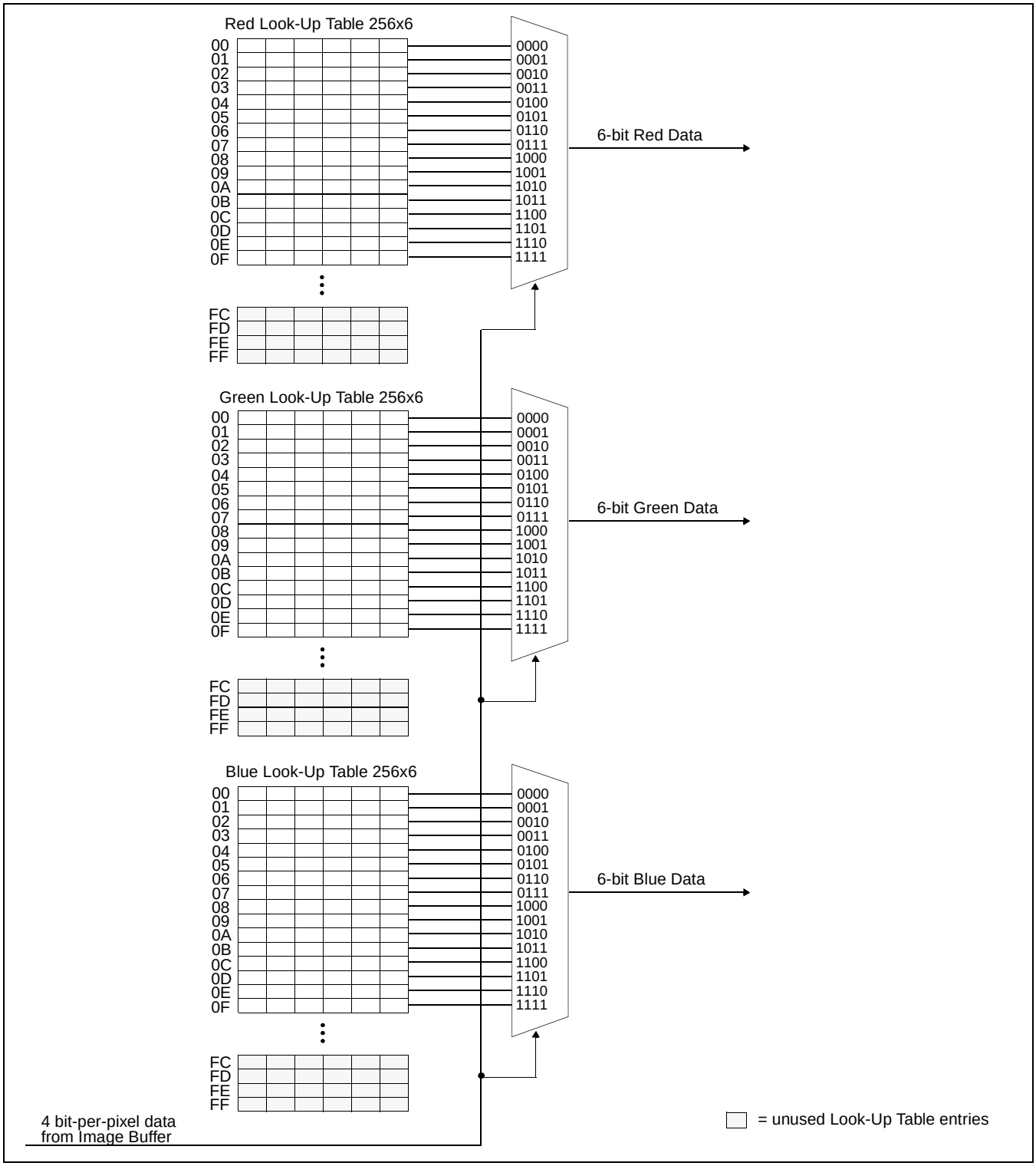


Figure 11-7 4 Bit-Per-Pixel Color Mode Data Output Path

8 Bit-per-pixel Color Mode

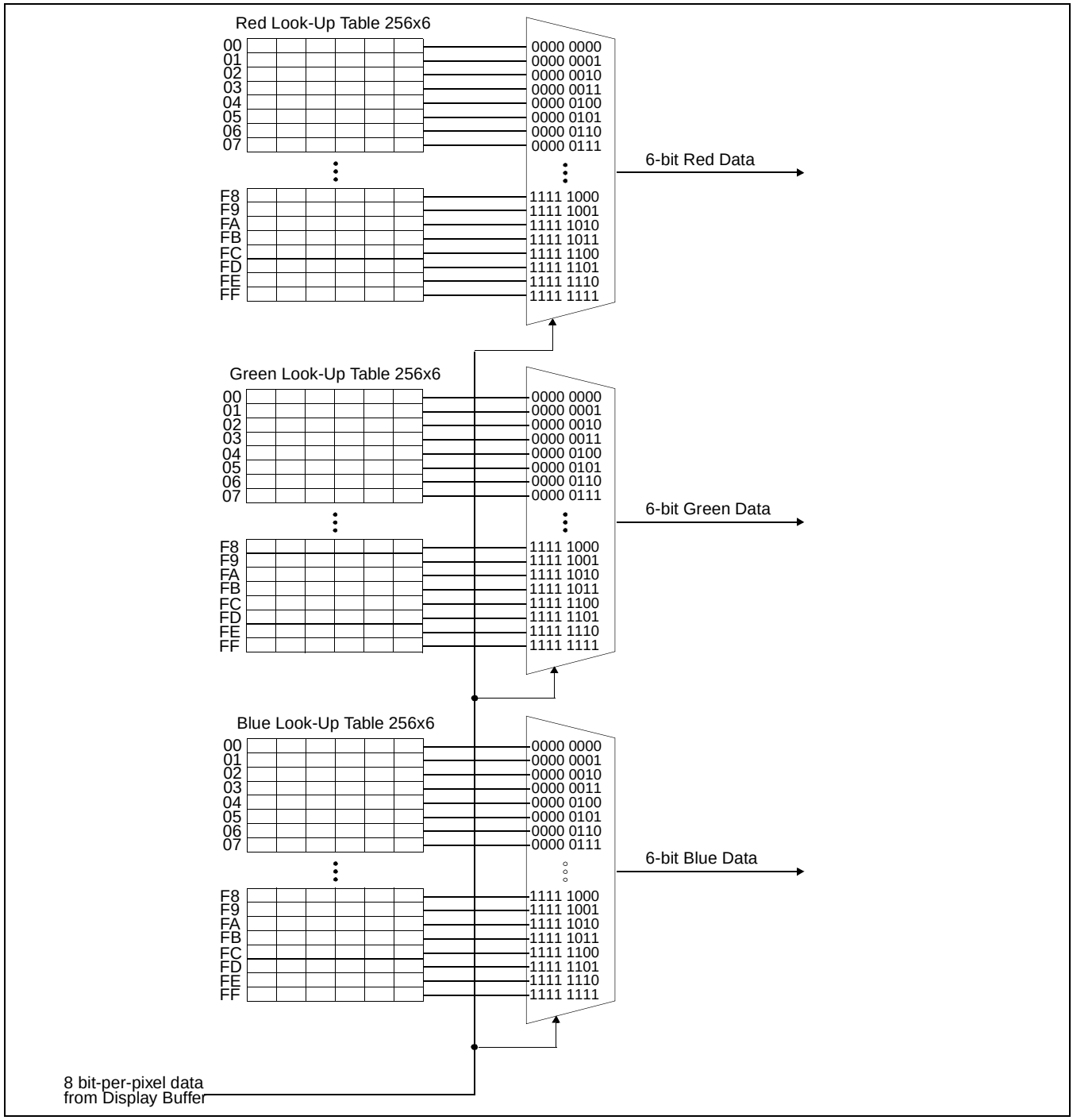


Figure 11-8 8 Bit-per-pixel Color Mode Data Output Path

16 Bit-Per-Pixel Color Mode

The LUT is bypassed and the color data is directly mapped for this color depth– See “Display Data Formats” on page 172.

12 SwivelView™

12.1 Concept

Most computer displays are refreshed in landscape orientation – from left to right and top to bottom. Computer images are stored in the same manner. SwivelView™ is designed to rotate the displayed image on an LCD by 90°, 180°, or 270° in an counter-clockwise direction. The rotation is done in hardware and is transparent to the user for all display buffer reads and writes. By processing the rotation in hardware, SwivelView™ offers a performance advantage over software rotation of the displayed image.

The image is not actually rotated in the display buffer since there is no address translation during CPU read/write. The image is rotated during display refresh.

12.2 90° SwivelView™

90° SwivelView™ requires the Memory Clock (MCLK) to be at least 1.25 times the frequency of the Pixel Clock (PCLK), i.e. MCLK ≥ 1.25PCLK.

The following figure shows how the programmer sees a 320x480 portrait image and how the image is being displayed. The application image is written to the S1D13708 in the following sense: A–B–C–D. The display is refreshed by the S1D13708 in the following sense: B-D-A-C.

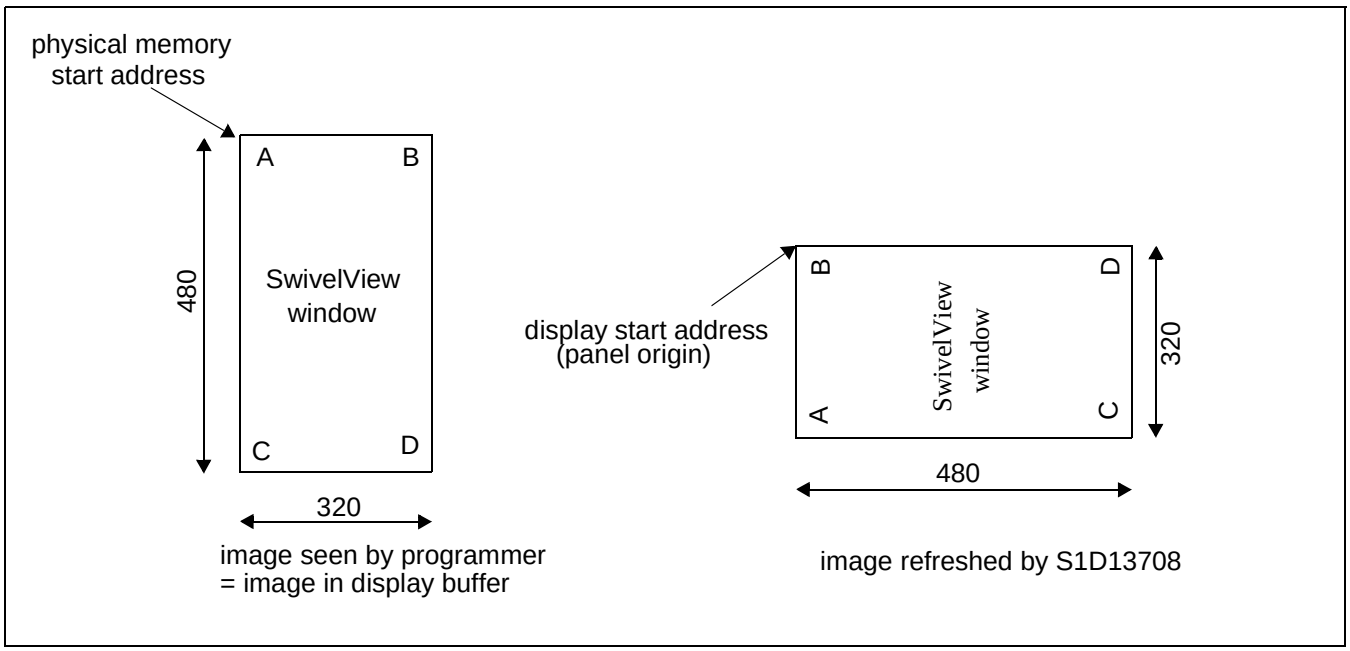


Figure 12-1 Relationship Between The Screen Image and the Image Refreshed in 90° SwivelView.

12.2.1 Register Programming

Enable 90° SwivelView™ Mode

Set SwivelView™ Mode Select bits to 01.

Display Start Address

The display refresh circuitry starts at pixel “B”, therefore the Display Start Address register must be programmed with the address of pixel “B”. The example in the figure shows a 320-pixel wide display, and if we assume 8-bpp display mode, the Display Start Address will be 4Fh (the Display Start Address register is 0-based and in 32-bit increment).

Memory Address Offset

The Memory Address Offset register should be normally set to be the same as the display width, e.g. 320 pixels or 50h (the Memory Address Offset register is in 32-bit increment). This value may be increased to create a virtual display.

Panning

Panning is achieved by changing the Display Start Address register:

- Increment/decrement the Display Start Address register pans the display window right/left by 32 bits, e.g. 4 pixels in 8-bpp mode.
- Increase/decrease the Display Start Address register by an amount equals to the Memory Address Offset pans the display window down/up by 1 line.

12.3 180° SwivelView™

The following figure shows how the programmer sees a 480x320 landscape image and how the image is being displayed. The application image is written to the S1D13708 in the following sense: A–B–C–D. The display is refreshed by the S1D13708 in the following sense: D-C-B-A.

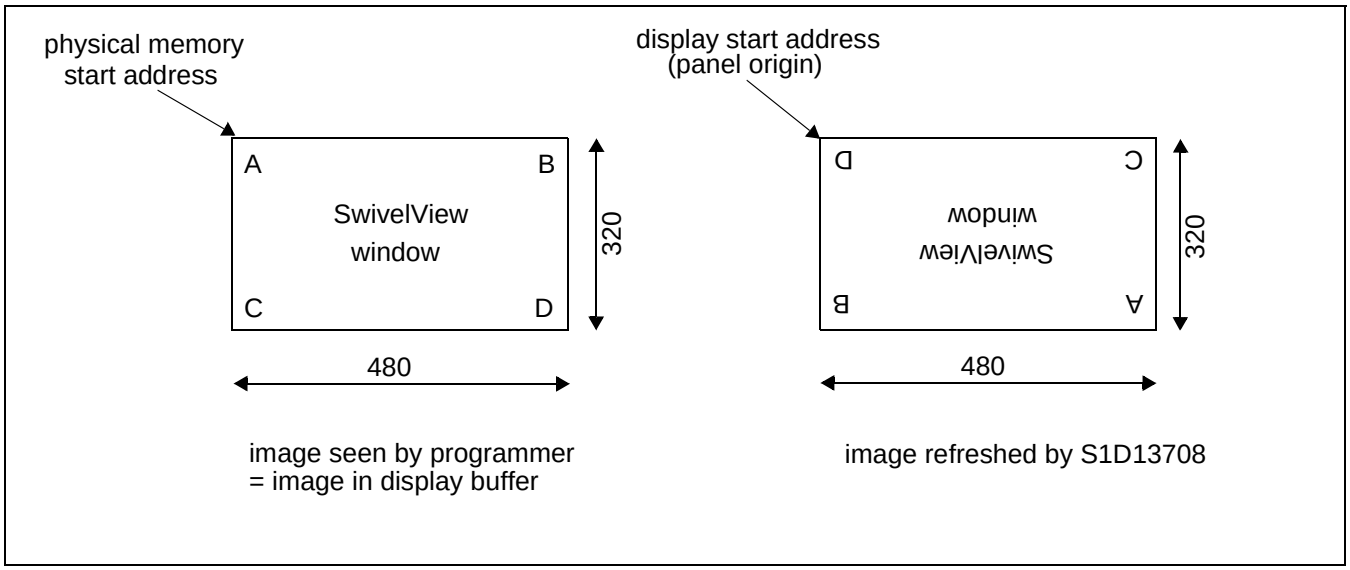


Figure 12-2 Relationship Between The Screen Image and the Image Refreshed in 180° SwivelView.

12.3.1 Register Programming

Enable 180° SwivelView™ Mode

Set SwivelView™ Mode Select bits to 10.

Display Start Address

The display refresh circuitry starts at pixel “D”, therefore the Display Start Address register must be programmed with the address of pixel “D”. The example in the figure shows a 480-pixel wide display, and if we assume 8-bpp display mode, the Display Start Address will be 95FFh (the Display Start Address register is 0-based and in 32-bit increment).

Memory Address Offset

The Memory Address Offset register should be normally set to be the same as the display width, e.g. 480 pixels or 78h (the Memory Address Offset register is in 32-bit increment). This value may be increased to create a virtual display.

Panning

Panning is achieved by changing the Display Start Address register:

- Increment/decrement the Display Start Address register pans the display window right/left by 32 bits, e.g. 4 pixels in 8-bpp mode.
- Increase/decrease the Display Start Address register by an amount equals to the Memory Address Offset pans the display window down/up by 1 line.

12.4 270° SwivelView™

270° SwivelView™ requires the Memory Clock (MCLK) to be at least 1.25 times the frequency of the Pixel Clock (PCLK), i.e. MCLK ≥ 1.25PCLK.

The following figure shows how the programmer sees a 320x480 portrait image and how the image is being displayed. The application image is written to the S1D13708 in the following sense: A–B–C–D. The display is refreshed by the S1D13708 in the following sense: C-A-D-B.

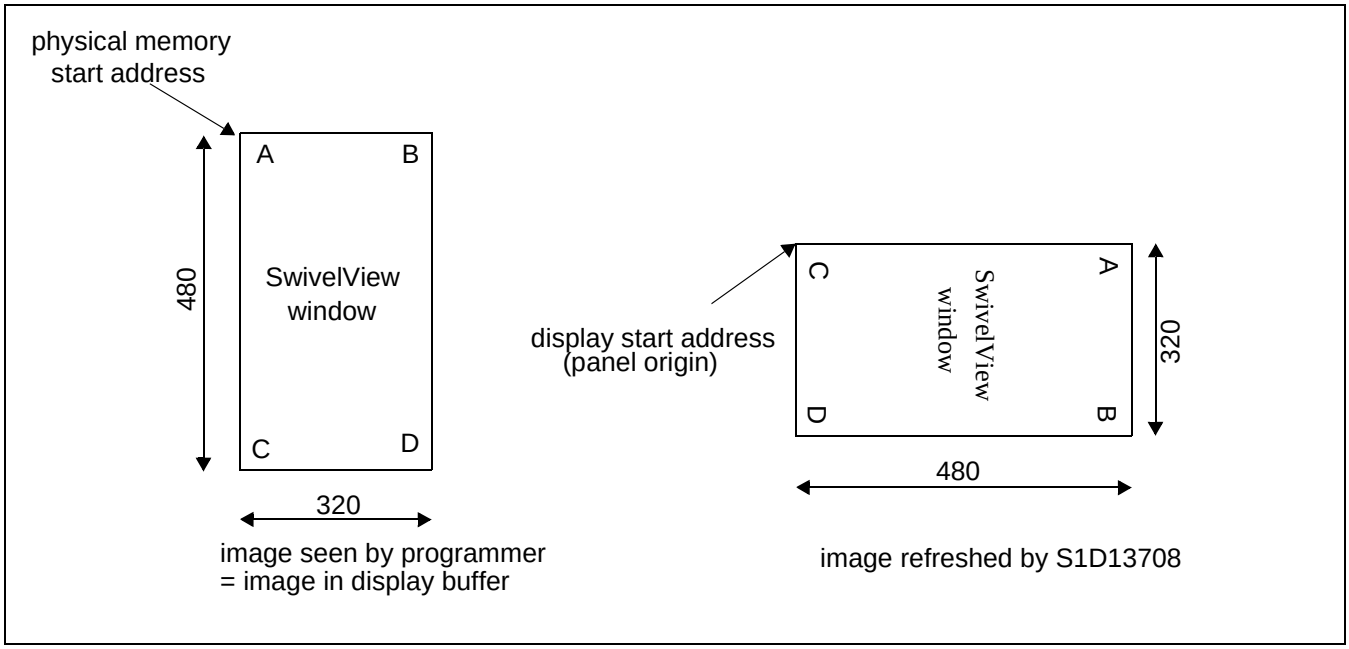


Figure 12-3 Relationship Between The Screen Image and the Image Refreshed in 270° SwivelView.

12.4.1 Register Programming

Enable 270° SwivelView™ Mode

Set SwivelView™ Mode Select bits to 11.

Display Start Address

The display refresh circuitry starts at pixel “C”, therefore the Display Start Address register must be programmed with the address of pixel “C”. The example in the figure shows a 320-pixel wide display, and if we assume 8-bpp display mode, the Display Start Address will be 95B0h (the Display Start Address register is 0-based and in 32-bit increment).

Memory Address Offset

The Memory Address Offset register should be normally set to be the same as the display width, e.g. 320 pixels or 50h (the Memory Address Offset register is in 32-bit increment). This value may be increased to create a virtual display.

Panning

Panning is achieved by changing the Display Start Address register:

- Increment/decrement the Display Start Address register pans the display window right/left by 32 bits, e.g. 4 pixels in 8-bpp mode.
- Increase/decrease the Display Start Address register by an amount equals to the Memory Address Offset pans the display window down/up by 1 line.

13 Picture-in-Picture Plus (PIP+)

13.1 Concept

Picture-in-Picture Plus enables a secondary window (or PIP+ window) within the main display window. The PIP+ window may be positioned anywhere within the virtual display and is controlled through the PIP+ Window control registers (REG[7Ch] through REG[91h]). The PIP+ window retains the same color depth and SwivelView orientation as the main window.

The following diagram shows an example of a PIP+ window within a main window and the registers used to position it.

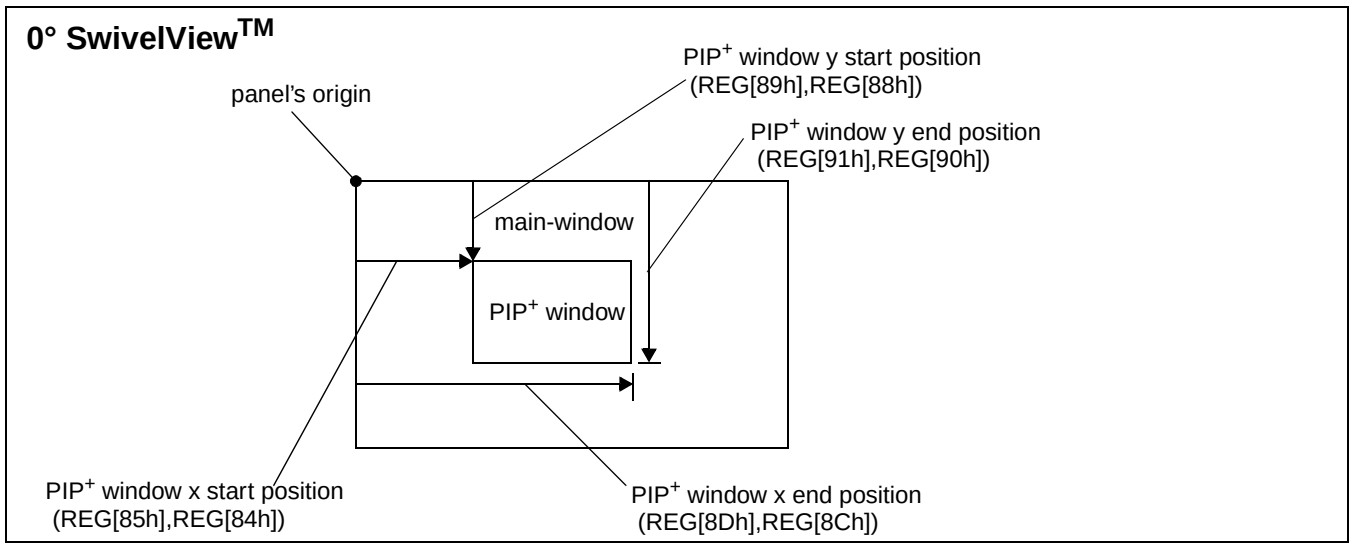


Figure 13-1 Picture-in-Picture Plus with SwivelView disabled

13.2 With SwivelView Enabled

13.2.1 SwivelView 90°

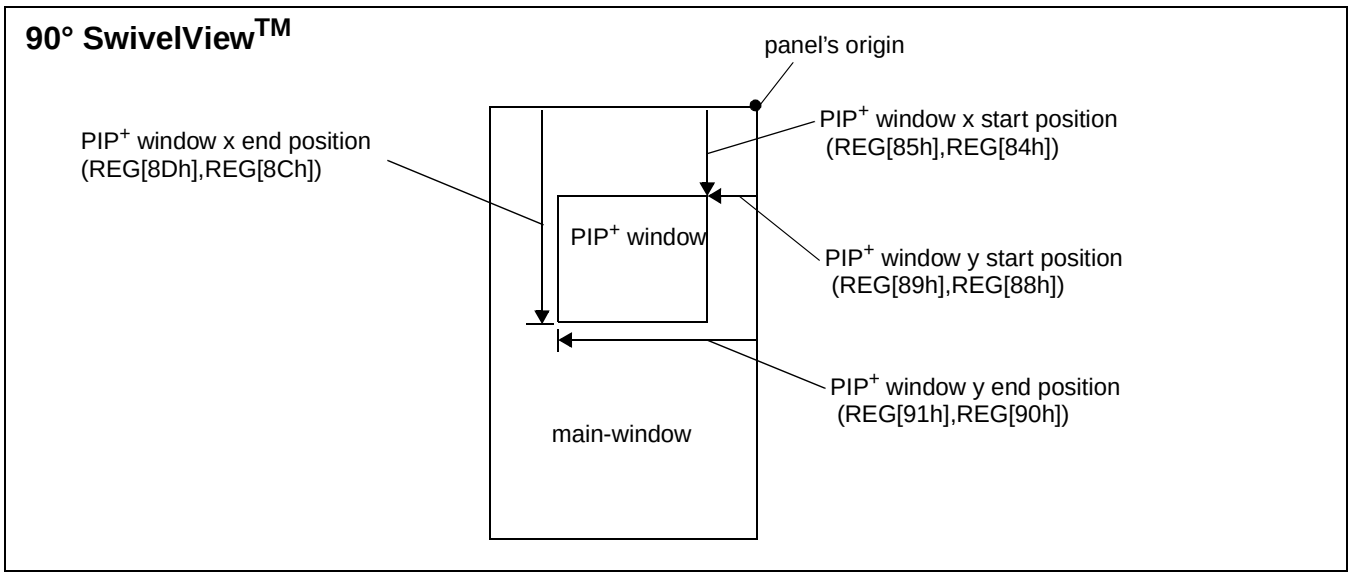


Figure 13-2 Picture-in-Picture Plus with SwivelView 90° enabled

13.2.2 SwivelView 180°

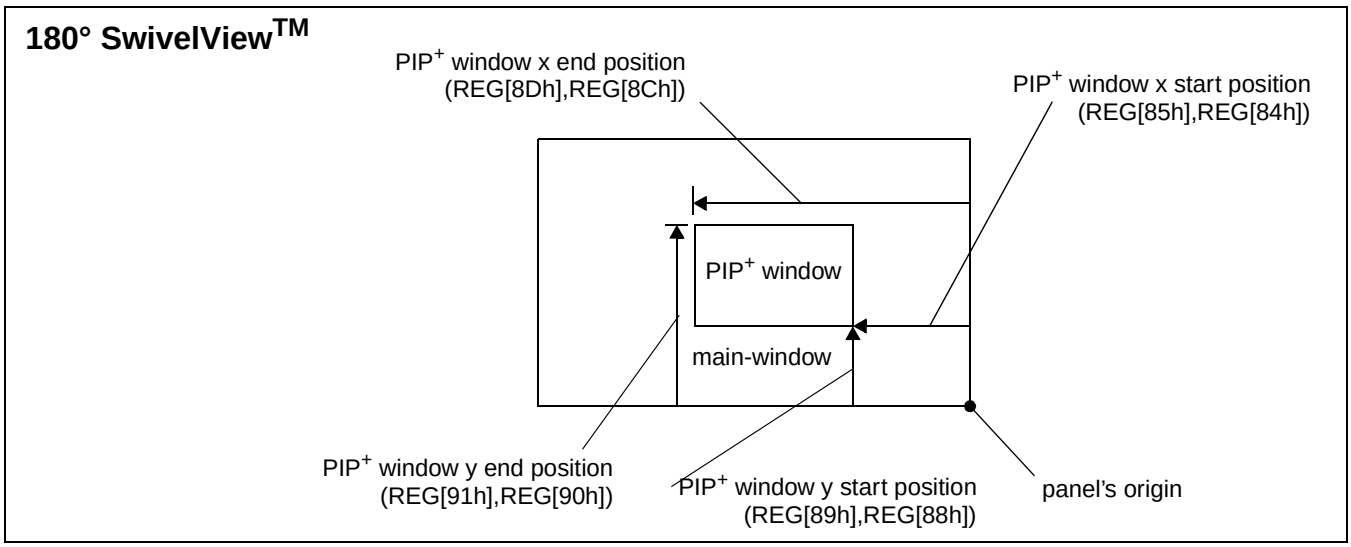


Figure 13-3 Picture-in-Picture Plus with SwivelView 180° enabled

13.2.3 SwivelView 270°

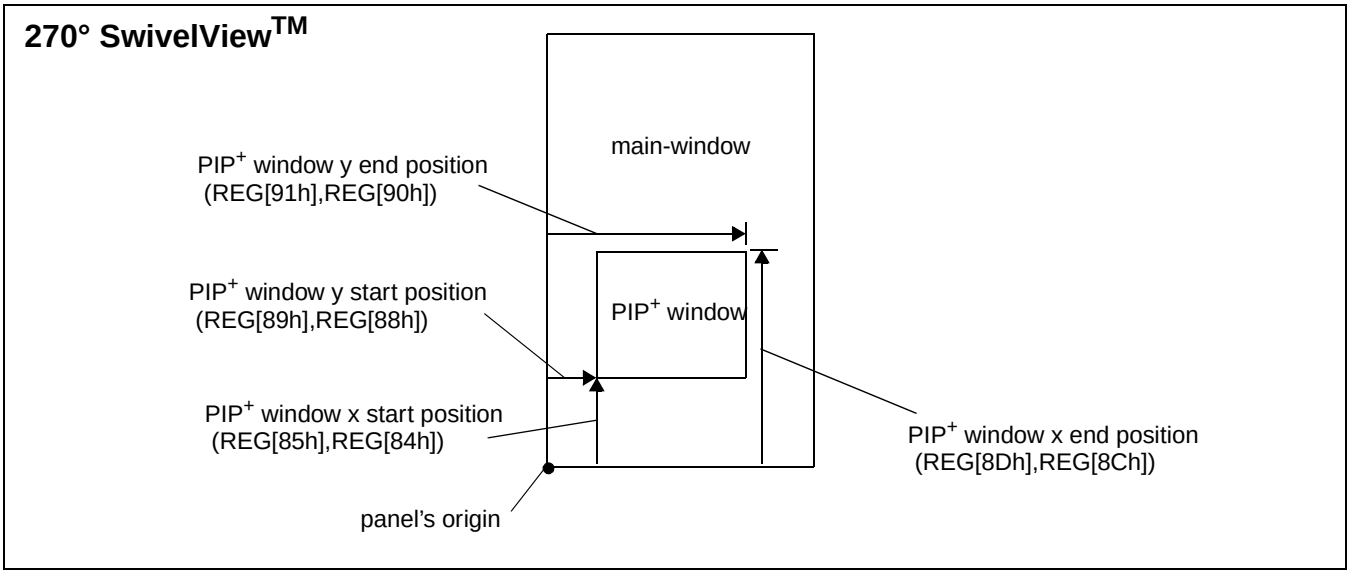


Figure 13-4 Picture-in-Picture Plus with SwivelView 270° enabled

14 Ink Layer

The S1D13708 Ink Layer design provides support for a foreground image that can be overlaid on the background (or main) image. The Ink Layer supports all color depths and automatically adjusts to the resolution of the display.

14.1 Memory Mapping

The S1D13708 has 80K bytes of embedded SRAM. When the ink layer is enabled, this memory is divided into two 40K byte blocks which can be logically represented as shown below.

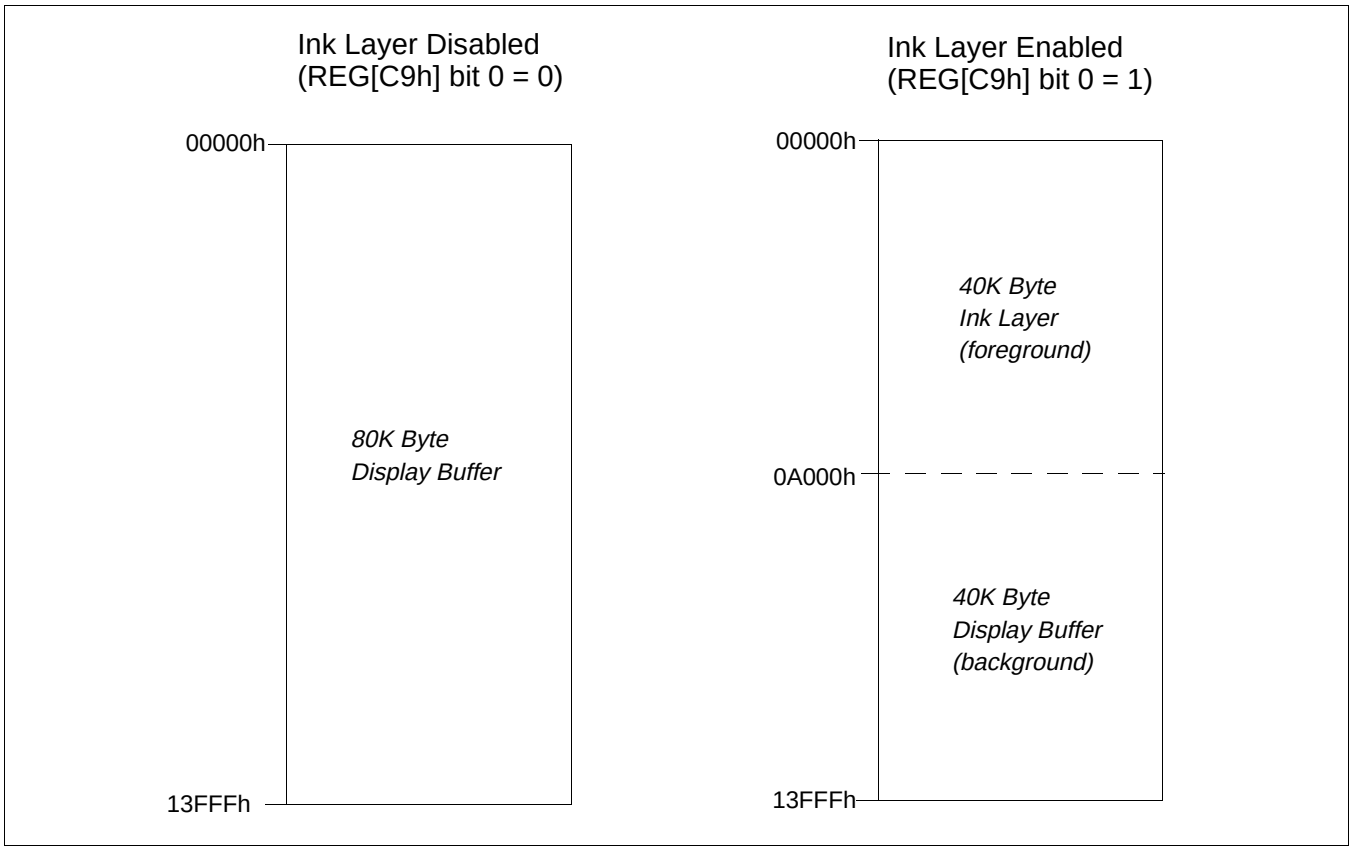


Figure 14-1 Memory Mapping for Ink Layer

14.2 Controlling the Ink Layer

The Ink Layer is controlled using REG[C7h] through REG[C9h].

Enabling the Ink Layer

The Ink Layer is enabled/disabled using REG[C9h] bit 0. When the Ink Layer Enable bit is set to 1, the display buffer is automatically configured for a foreground and background image as shown in Figure 14-1 Memory Mapping for Ink Layer.

Setting the Transparent Color

The Ink Layer requires a transparent color to be set. This transparent color is stored in REG[C7h], REG[C8h] and the value from these registers is compared with the values in the foreground image during display refresh. If the pixel value matches the transparent color, the corresponding pixel from the background image is shown. If the pixel value does not match the transparent color, the foreground image is shown.

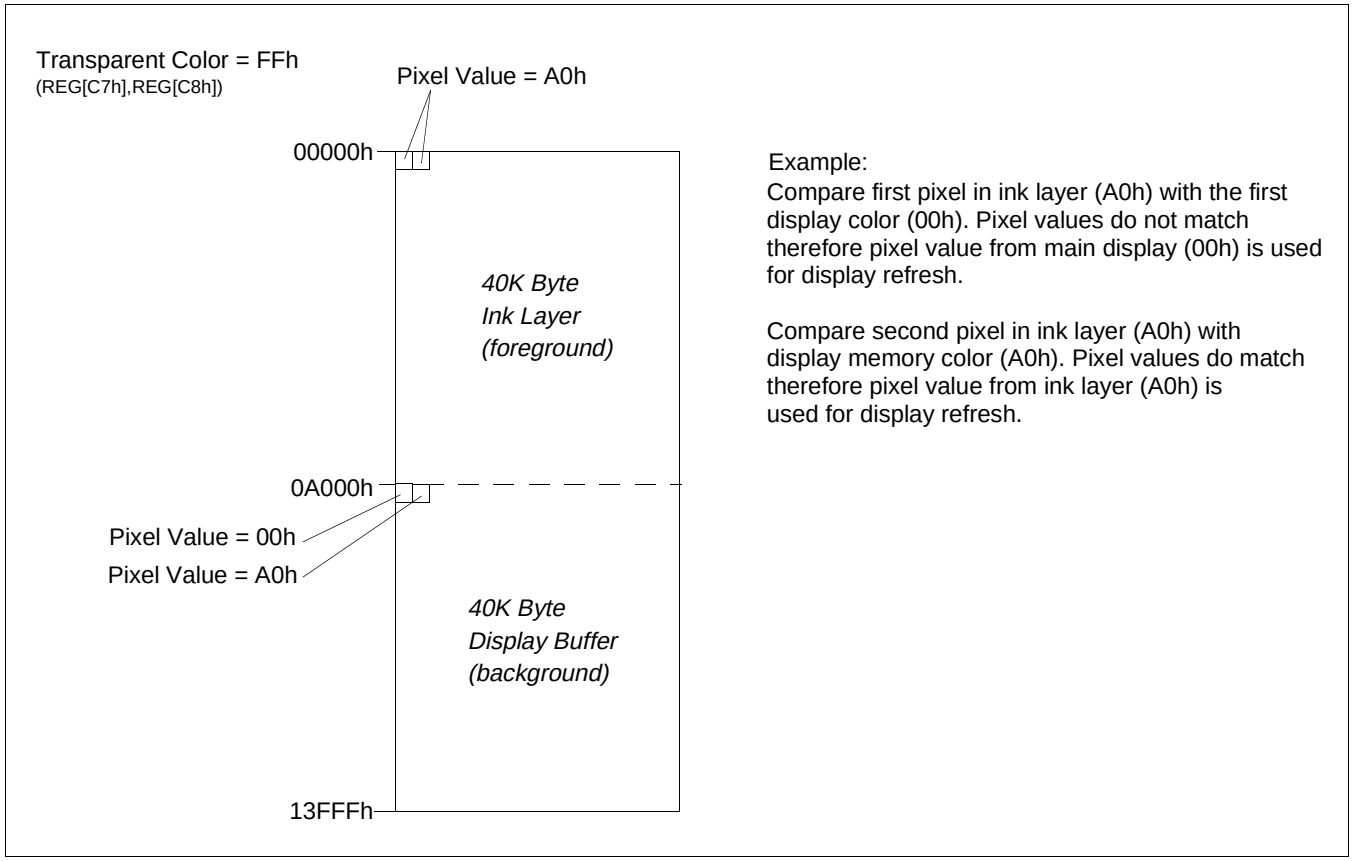


Figure 14-2 Transparent Color Example

14.3 Limitations

There are certain limitations when the Ink Layer is enabled.

- Available display buffer is only 40K byte. This limits the resolution and color depth of the display (e.g. 160x160 at 16 bpp is not possible).

15 Indirect Interface

The Indirect Interface is an asynchronous interface with 2 modes of operation, mode 68 and mode 80. The address and data are multiplexed onto the data bus, thus providing a low pin count interface to the S1D13708. The modes are distinguished between one another by the polarity of BS#. Both modes support 8 and 16 bit accesses for both little and big endian.

The S1D13708 Indirect Interface use a combination of “command” and “data” reads/writes to program the LCD controller. First, register addresses are loaded using “command” writes. Then, register values and memory reads/writes are performed using “data” reads/writes. Memory addresses are performed by programming REG[C0h] through REG[C2h] with the desired 17-bit address.

If the Memory Access Select bit is enabled (REG[C6h] bit 0 = 1), memory accesses are always word accesses and the signals WRU#, RDU# and EBU are ignored (WRL#, RDL# and EBL are used for accessing both the higher and lower bytes). If this bit is disabled, byte/word accesses to memory are controlled by WRn#, RDn# and EBn.

For details and examples, see the following sections. Also refer to *Connecting to the Indirect Interface Bus*, Document Number X39A-G-020-xx.

15.1 Mode 68

The following shows an example of a “register write” with Mode 68.

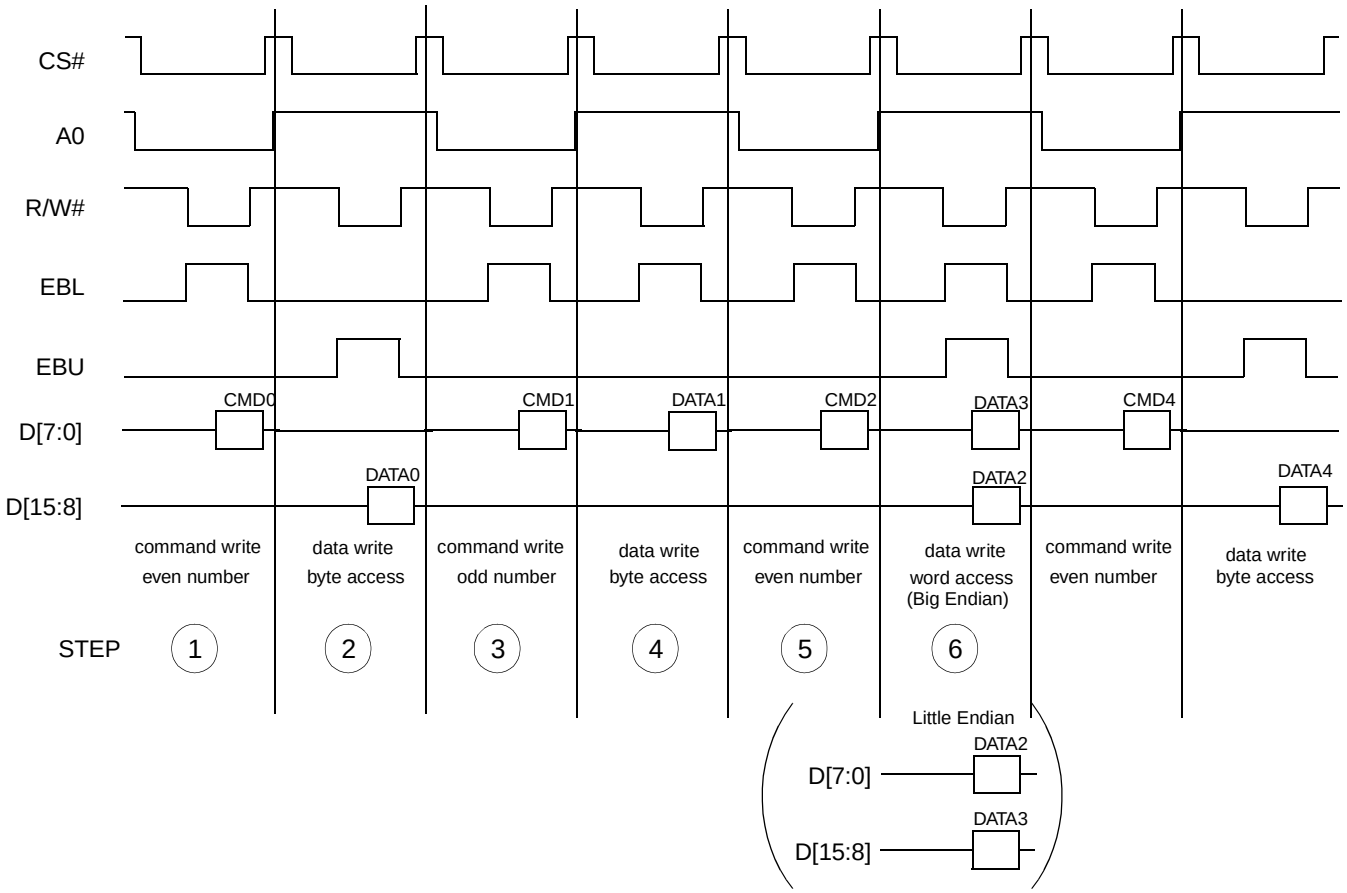


Figure 15-1 Sample timing of “register write” with Mode 68

1. write register address (command write).
2. write register data (data write). Even numbered register uses the high byte.
3. write register address (command write).
4. write register data (data write). Demonstrates how to access an odd numbered register using the low byte. Note that the high byte could also have been used by asserting EBU instead of EBL.
5. write register address (command write)
6. write register data (data write). Word accesses (16-bit) use the higher byte for the even register address and the lower byte for the odd register address.

The following shows an example of a “register read” with Mode 68.

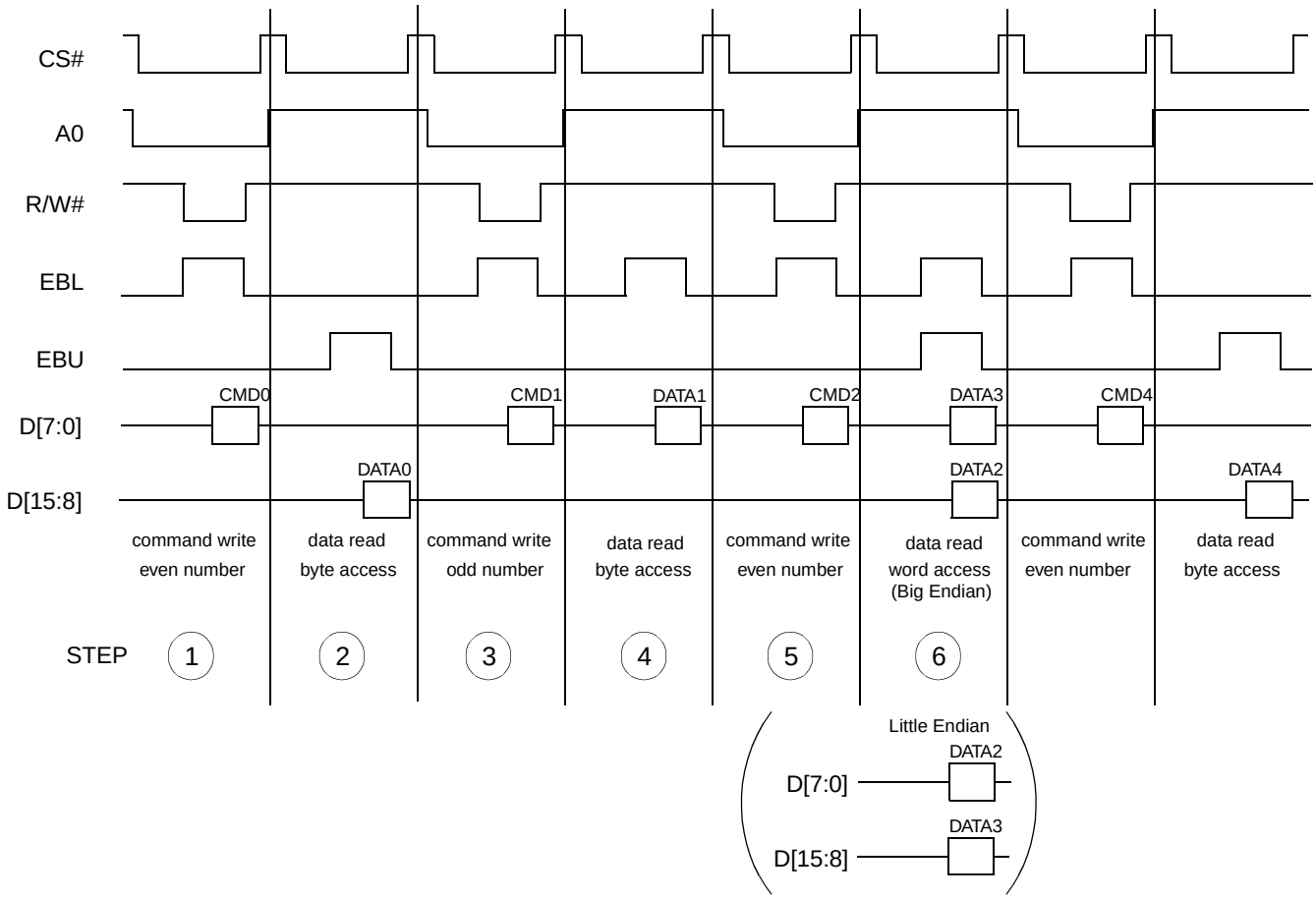


Figure 15-2 Sample timing of “register read” with Mode 68

1. write register address (command write).
2. read register data (data read).
3. write register address (command write).
4. read register data (data read). Demonstrates how to access an odd numbered register using the low byte. Note that the high byte could also have been used by asserting EBU instead of EBL.
5. write register address (command write).
6. read register data (data read). Word accesses (16-bit) use the higher byte for the even register address and the lower byte for the odd register address.

The following shows an example of a “memory write” with Mode 68, Big Endian.

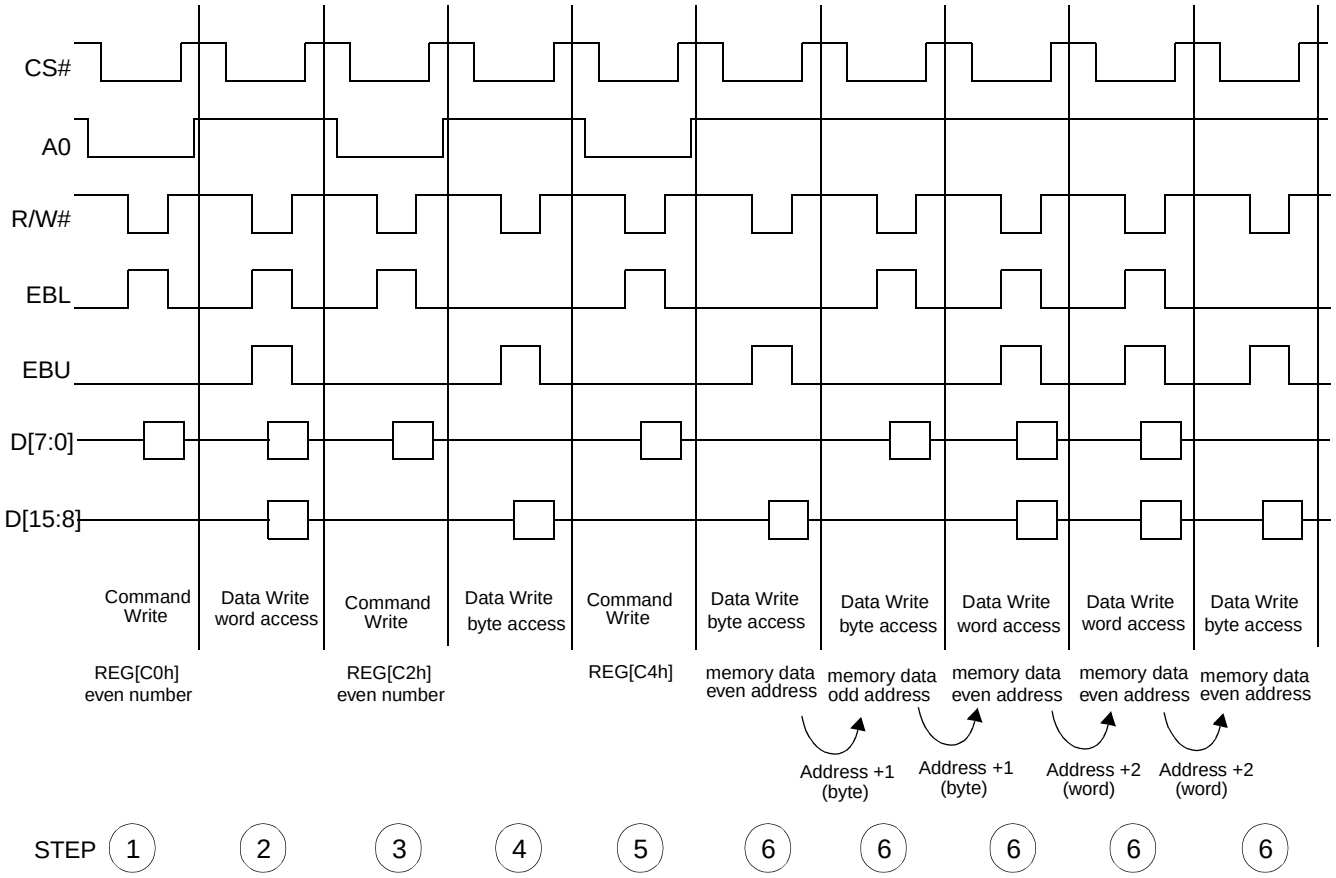


Figure 15-3 Sample timing of “memory write” with Mode 68, Big Endian

1. write register address of Memory Access Pointer 0 (REG[C0h]) (command write).
2. write memory address[7:0] to the low byte and memory address[15:8] to the high byte (data write). This places data into RegC0 and RegC1, which form bits [7:0] and [15:8] of memory address respectfully.
3. write register address of Memory Access Pointer 2 (REG[C2h] bit 0) (command write).
4. write memory access pointer (REG[c2h] bit 0) (data write). This forms bit 16 of memory address.
5. write register number of Memory Access Start register (REG[C4h] (command write).

Note

No “data write” is required after a command write to the Memory Access Start register (REG[C4h]). This step configures the S1D13708 for burst memory access beginning with the next data write.

6. write Memory data (data write)
   The S1D13708 indirect interface implements an auto increment function to allow burst memory accesses. For byte accesses, the Memory Address Pointer registers (REG[C0h], REG[C1h], REG[C2h]) are automatically incremented “+1”. For word accesses, the Memory Address Pointer registers are automatically incremented “+2”.

The following shows an example of a “memory read” with Mode 68, Big Endian.

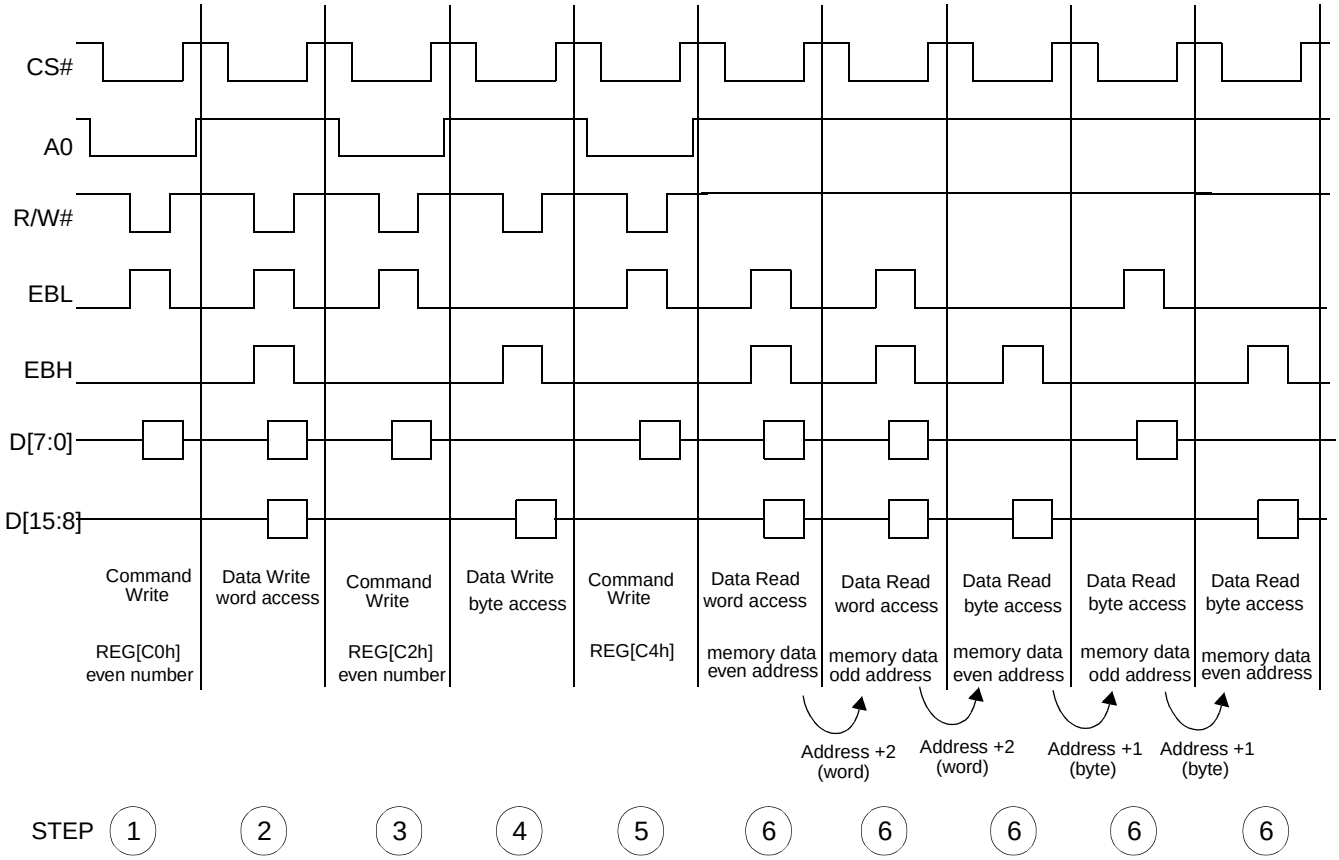


Figure 15-4 Sample timing of “memory read” with Mode 68, Big Endian

1. write register address of Memory Access Pointer 0 (REG[C0h]) (command write).
2. write memory address[7:0] to the low byte and memory address[15:8] to the high byte (data write). This places data into RegC0 and RegC1, which form bits [7:0] and [15:8] of memory address respectfully.
3. write register address of Memory Access Pointer 2 (REG[C2h] bit 0) (command write).
4. write memory access pointer (REG[c2h] bit 0) (data write). This forms bit 16 of memory address.
5. write register number of Memory Access Start register (REG[C4h] (command write).

Note

No “data write” is required after a command write to the Memory Access Start register (REG[C4h]). This step configures the S1D13708 for burst memory access beginning with the next data write.

6. read Memory data (data read)
   The S1D13708 indirect interface implements an auto increment function to allow burst memory accesses. For byte accesses, the Memory Address Pointer registers (REG[C0h], REG[C1h], REG[C2h]) are automatically incremented “+1”. For word accesses, the Memory Address Pointer registers are automatically incremented “+2”.

The following shows an example of a “memory write” for Mode 68 when the Memory Access Select bit is enabled (REG[C6h] bit 0 = 1).

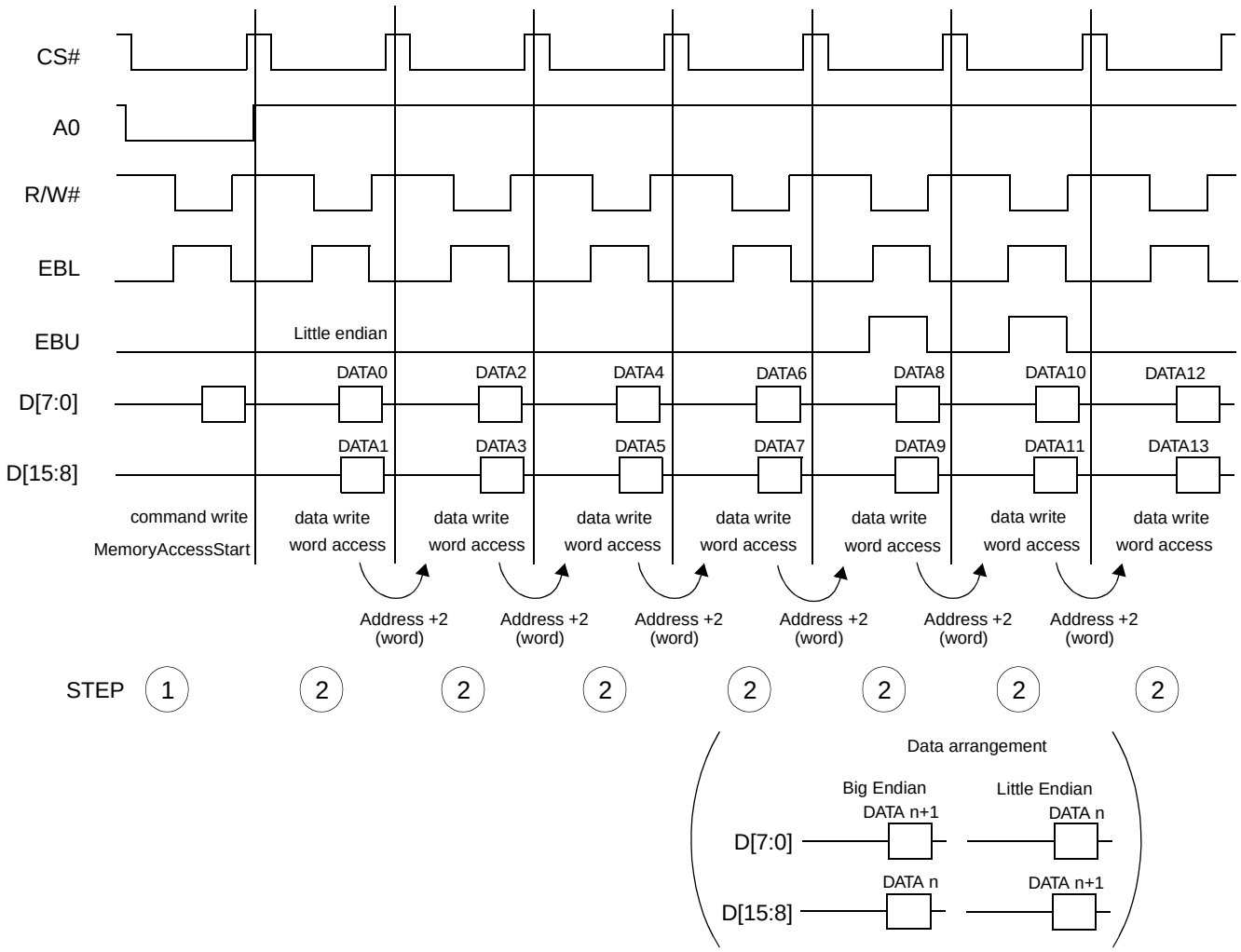


Figure 15-5 Sample timing of “register write” for Mode 68 when Memory Access Select Enabled

1. write register number of Memory Access Start register (REG[C4h]) (command write).

Note

No “data write” is required after a command write to the Memory Access Start register (REG[C4h]). This step configures the S1D13708 for burst memory access beginning with the next data write.

2. write Memory data (data write).
   If the Memory Access Select bit (REG[C6h] bit 0 = 1), memory accesses are word accesses even if EBU is high (EBU is ignored and EBL is used to write both the upper and lower bytes). The big/little endian setting is used to determine the data arrangement for word accesses only.

The S1D13708 indirect interface implements an auto increment function to allow burst memory accesses. For byte accesses, the Memory Address Pointer registers (REG[C0h], REG[C1h], REG[C2h]) are automatically incremented “+1”. For word accesses, the Memory Address Pointer registers are automatically incremented “+2”.

Note

If the Memory Access Select bit is enabled (REG[C6h] bit 0 = 1), all memory accesses are word accesses (EBU is ignored). Therefore, the memory address set in REG[C0h] through REG[C2h] must be an even address.

The following shows an example of a “memory read” for Mode 68 when the Memory Access Select bit is enabled (REG[C6h] bit 0 = 1).

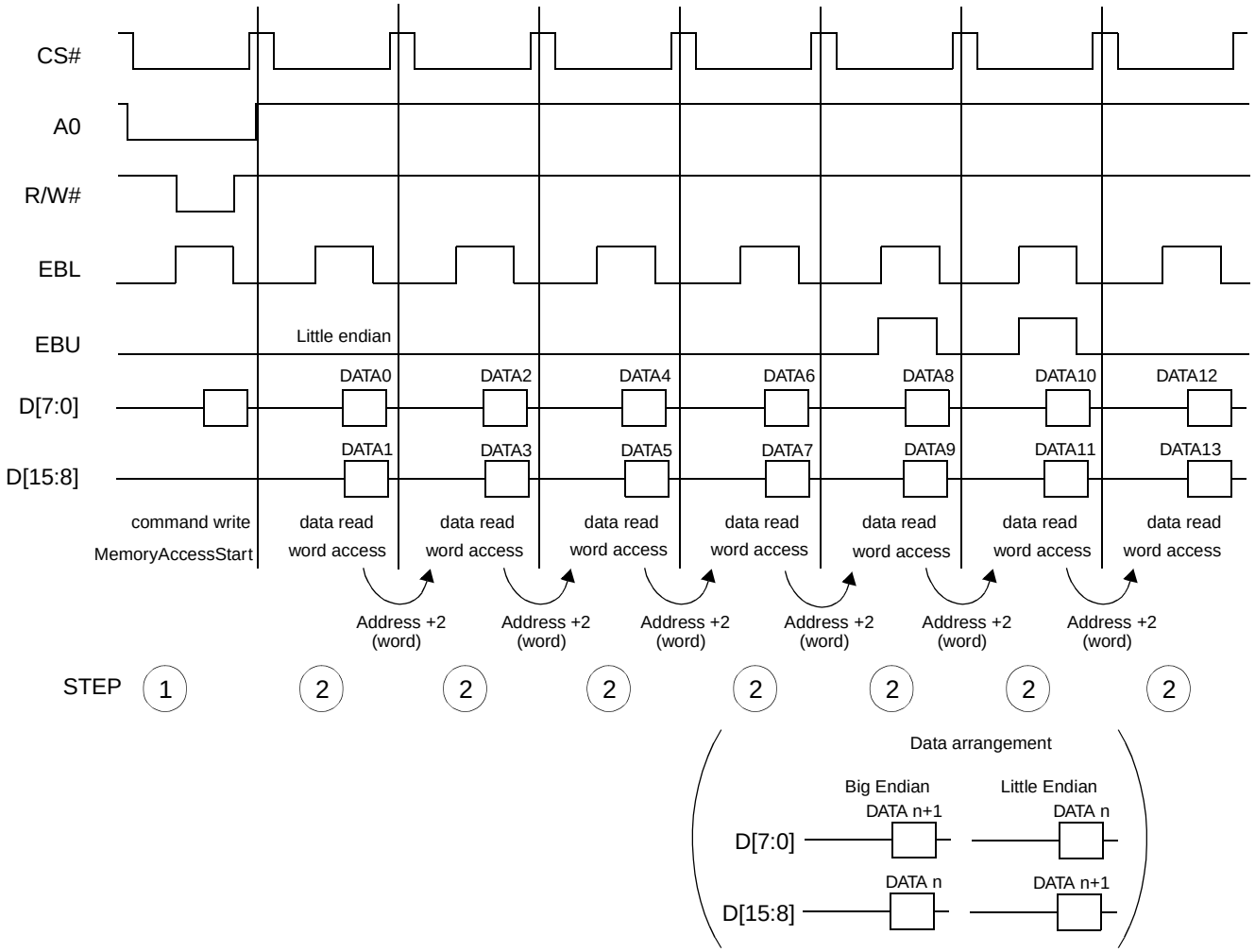


Figure 15-6 Sample timing of “register read” for Mode 68 when Memory Access Select Enabled

1. write register number of Memory Access Start register (REG[C4h]) (command write).

Note
No “data write” is required after a command write to the Memory Access Start register (REG[C4h]). This step configures the S1D13708 for burst memory access beginning with the next data read.

2. read Memory data (data read).
   If the Memory Access Select bit (REG[C6h] bit 0 = 1), memory accesses are word accesses even if EBU is high (EBU is ignored and EBL is used to read both the upper and lower bytes). The big/little endian setting is used to determine the data arrangement for word accesses only.

   The S1D13708 indirect interface implements an auto increment function to allow

burst memory accesses. For byte accesses, the Memory Address Pointer registers (REG[C0h], REG[C1h], REG[C2h]) are automatically incremented “+1”. For word accesses, the Memory Address Pointer registers are automatically incremented “+2”.

Note

If the Memory Access Select bit is enabled (REG[C6h] bit 0 = 1), all memory accesses are word accesses (EBU is ignored). Therefore, the memory address set in REG[C0h] through REG[C2h] must be an even address.

Register access still uses EBU irrespective of REG[C6] bit 0 value.

15.2 Mode 80

Mode 80 supports byte and word access for both register and memory access. It also allows both big and little endian modes. The following shows an example of a “register write” with Mode 80.

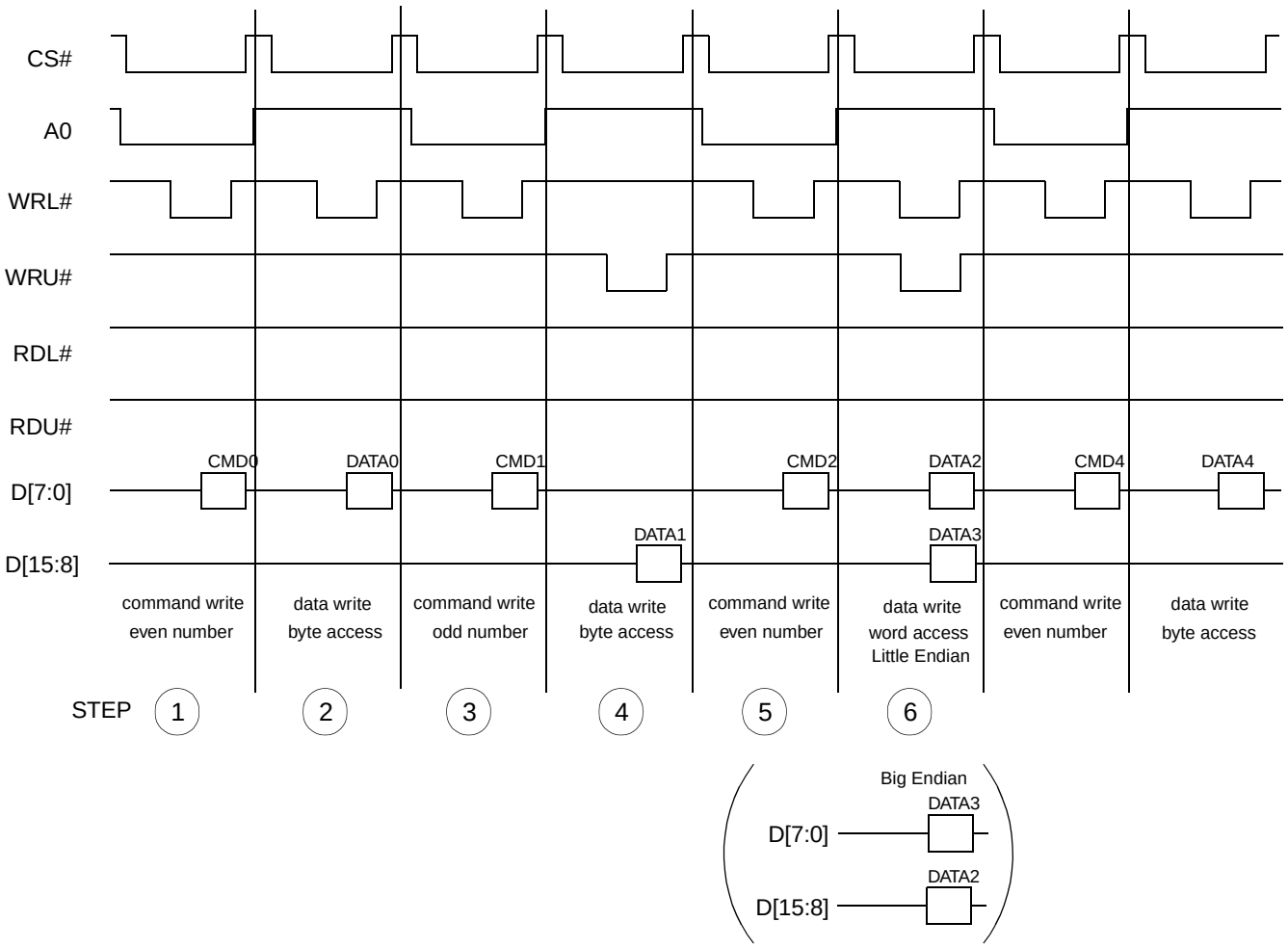


Figure 15-7 Sample timing of “register write” with Mode 80

1. write register address (command write). Command write is always the lower byte.
2. write register data (data write). Even numbered register uses the low byte.
3. write register address (command write).
4. write register data (data write). Demonstrates how to access an odd numbered register using the high byte. Note that the low byte could also have been used by asserting WRL# instead of WRU#.
5. write register address (command write)

6. write register data (data write). Word accesses (16-bit) use the lower byte for the lower register number and the higher byte for the higher register number.

The following shows an example of a “register read” with Mode 80.

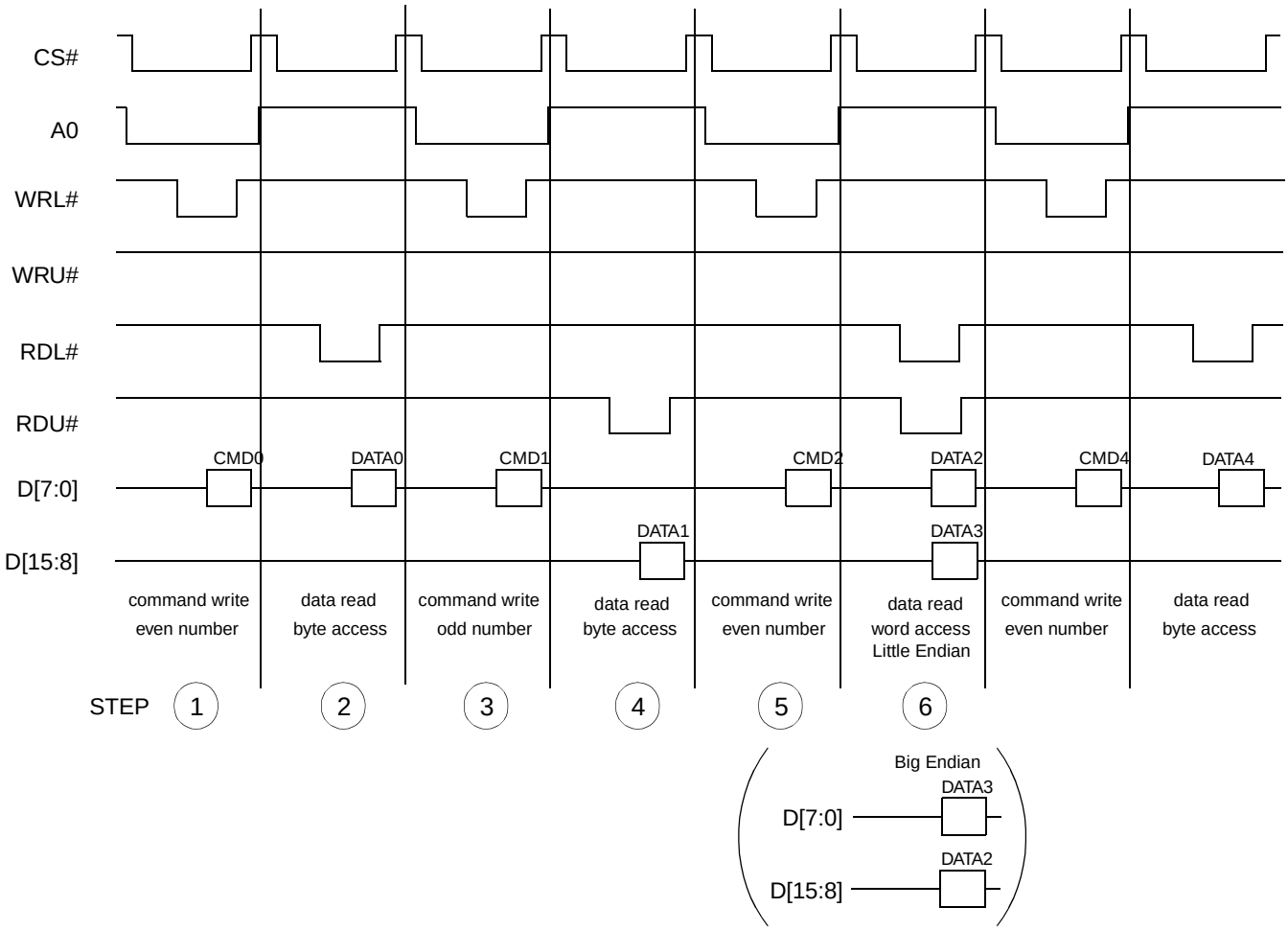


Figure 15-8 Sample timing of “register read” with Mode 80

1. write register address (command write). Command write is always the lower byte.
2. read register data (data read). Even numbered register uses the lower byte.
3. write register address (command write).
4. read register data (data read). Demonstrates how to access an odd numbered register using the high byte. Note that the low byte could also have been used by asserting WRL# instead of WRU#.
5. write register address (command write).
6. read register data (data read). Word accesses (16-bit) use the lower byte for the lower register number and the higher byte for the higher register number.

The following shows an example of a “memory write” with Mode 80, little endian.

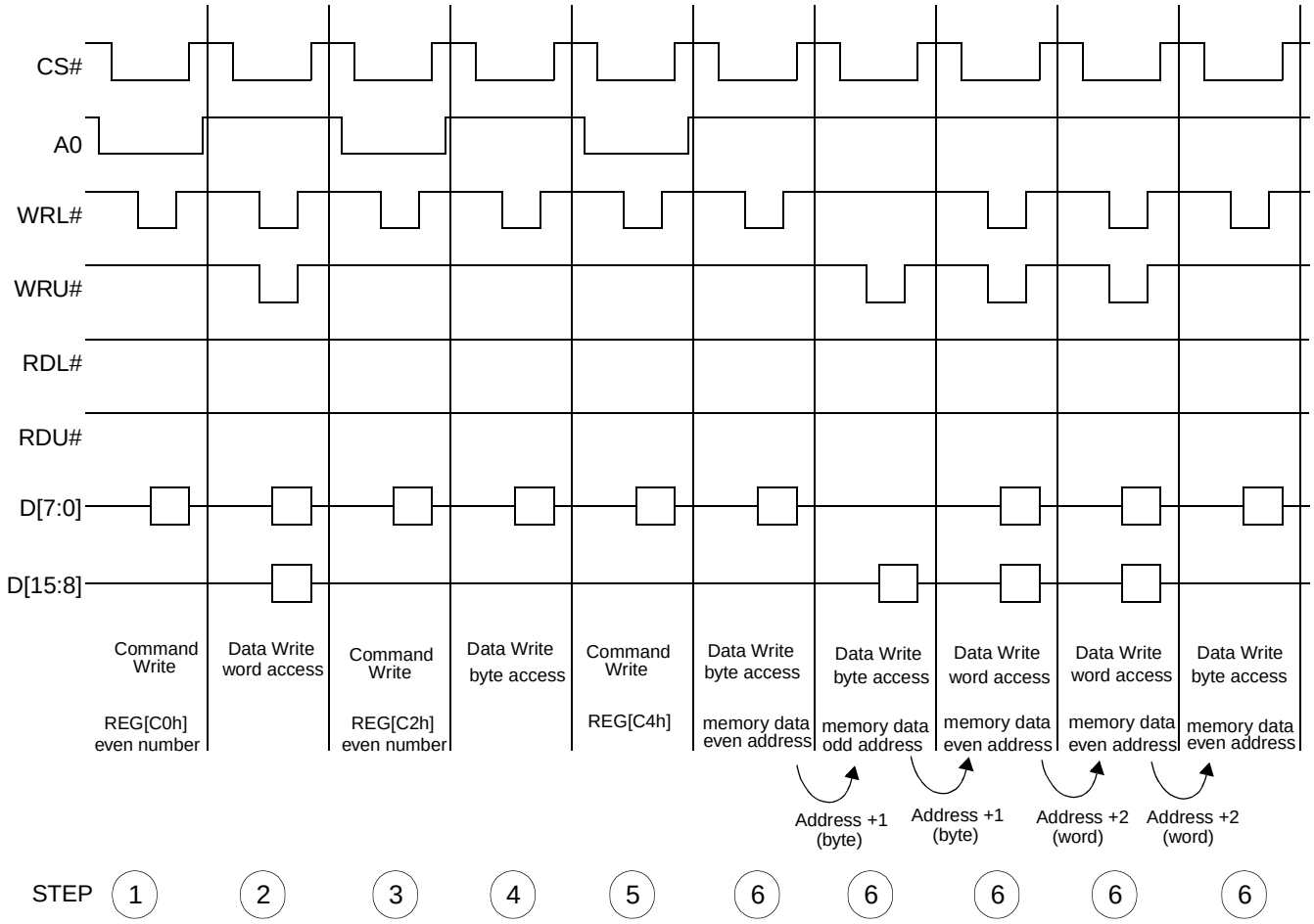


Figure 15-9 Sample timing of “memory write” with mode 80, little endian

1. write register number of Memory Access Pointer 0 (REG[C0h]) (command write).
2. write memory address 0 to the high byte and memory address 1 to the low byte (MA[15:0]) (data write).
3. write register number of Memory Access Pointer 2 (REG[C2h]) (command write).
4. write memory address 2 (MA16) to the low byte (data write).
5. write register number of Memory Access Start register (REG[C4h]) (command write).

Note

No “data write” is required after a command write to the Memory Access Start register (REG[C4h]). This step configures the S1D13708 for burst memory access beginning with the next data write.

6. write Memory data (data write)
   The S1D13708 indirect interface implements an auto increment function to allow burst memory accesses. For byte accesses, the Memory Address Pointer registers (REG[C0h], REG[C1h], REG[C2h]) are automatically incremented “+1”. For word accesses, the Memory Address Pointer registers are automatically incremented “+2”.

The following shows an example of a “memory read” with Mode 80, Little endian.

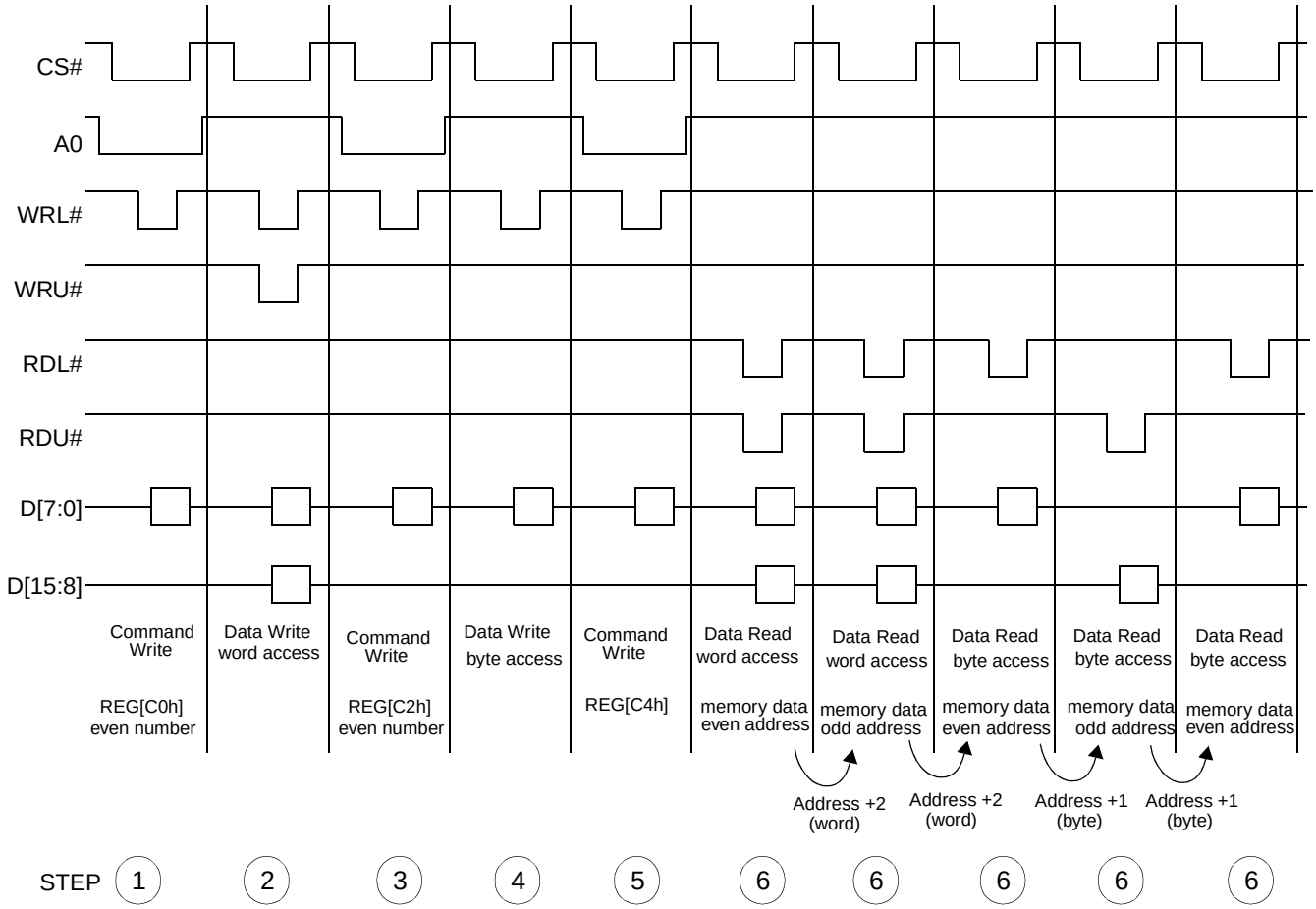


Figure 15-10 Sample timing of “memory read” with mode 80, Little endian

1. write register number of Memory Access Pointer 0 (REG[C0h]) (command write).
2. write memory address 0 to the high byte and memory address 1 to the low byte (MA[15:0]) (data write).
3. write register number of Memory Access Pointer 2 (REG[C2h]) (command write).
4. write memory address 2 (MA16) to the low byte (data write).
5. write register number of Memory Access Start register (REG[C4h]) (command write).

Note

No “data write” is required after a command write to the Memory Access Start register (REG[C4h]). This step configures the S1D13708 for burst memory access beginning with the next data write.

6. read Memory data (data read)
   The S1D13708 indirect interface implements an auto increment function to allow burst memory accesses. For byte accesses, the Memory Address Pointer registers (REG[C0h], REG[C1h], REG[C2h]) are automatically incremented “+1”. For word accesses, the Memory Address Pointer registers are automatically incremented “+2”.

The following shows an example of a “memory write” for Mode 80 when the Memory Access Select bit is enabled (REG[C6h] bit 0 = 1).

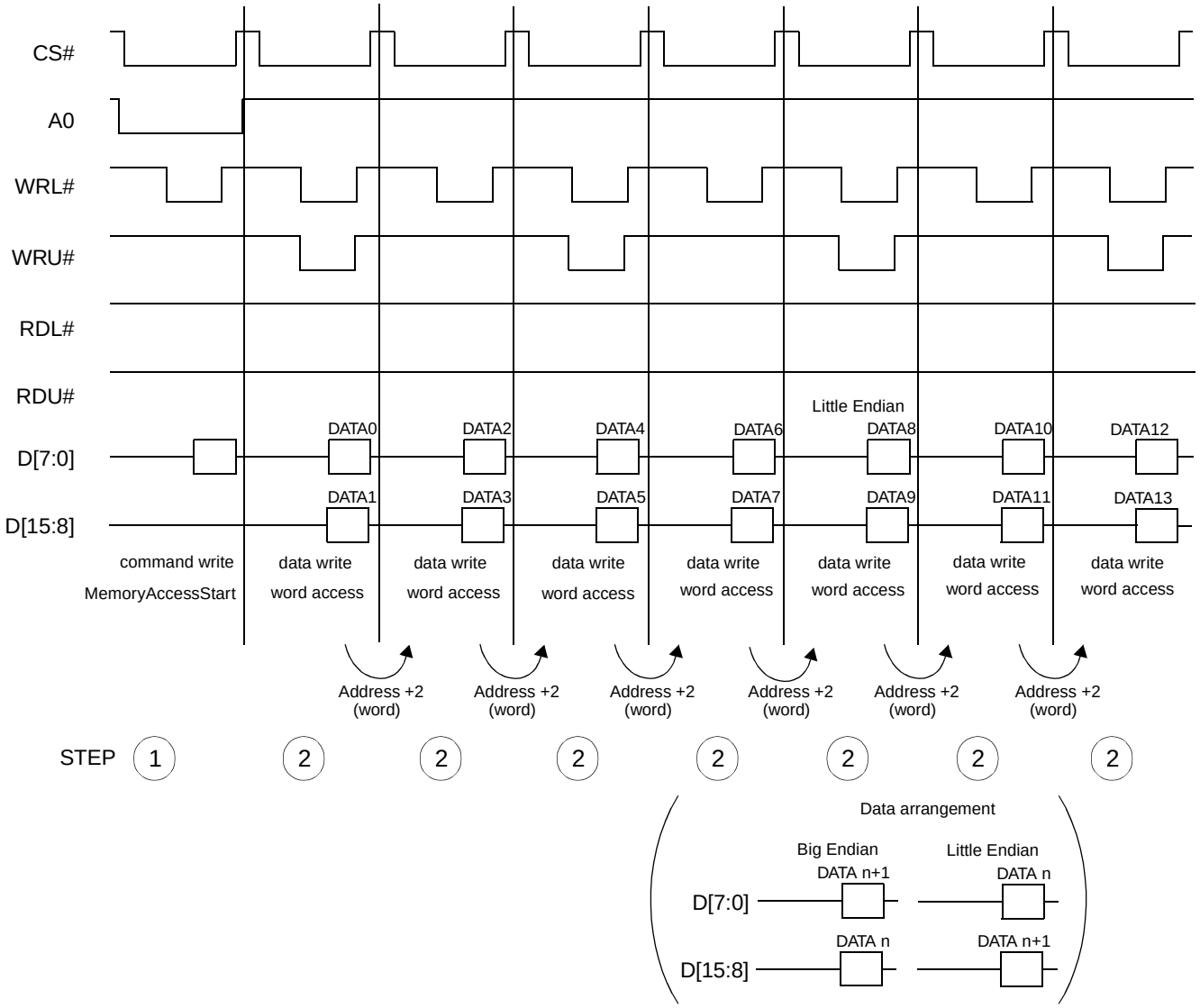


Figure 15-11 Sample timing of “memory write” for Mode 80 when Memory Access Select Enabled

1. write register address of Memory Access Start register (REG[C4h]) (command write).

Note

No “data write” is required after a command write to the memory Access Start register (REG[C4h]). This step configures the S1D13708 for burst memory access beginning with the next data write.

2. write Memory data (data write).
   If the Memory Access Select bit (REG[C6h] bit 0 = 1), memory accesses are word accesses even if WRU# is high (WRU# is ignored and WRL# is used to write both the

upper and lower bytes). The bit/Little endian setting is used to determine the data arrangement for word accesses only.

The S1D13708 indirect interface implements an auto increment function to allow burst memory accesses. For byte accesses, the Memory Address Pointer registers (REG[C0h], REG[C1h], REG[C2h]) are automatically incremented "+1". For word accesses, the Memory Address Pointer registers are automatically incremented "+2".

Note

If the Memory Access Select bit is enabled (REG[C6h] bit 0 = 1), all memory accesses are word accesses (WRU# is ignored). Therefore, the memory address set in REG[C0h] through REG[C2h] must be an even address.

The following shows an example of a “memory read” for Mode 80 when the Memory Access Select bit is enabled (REG[C6h] bit 0 = 1).

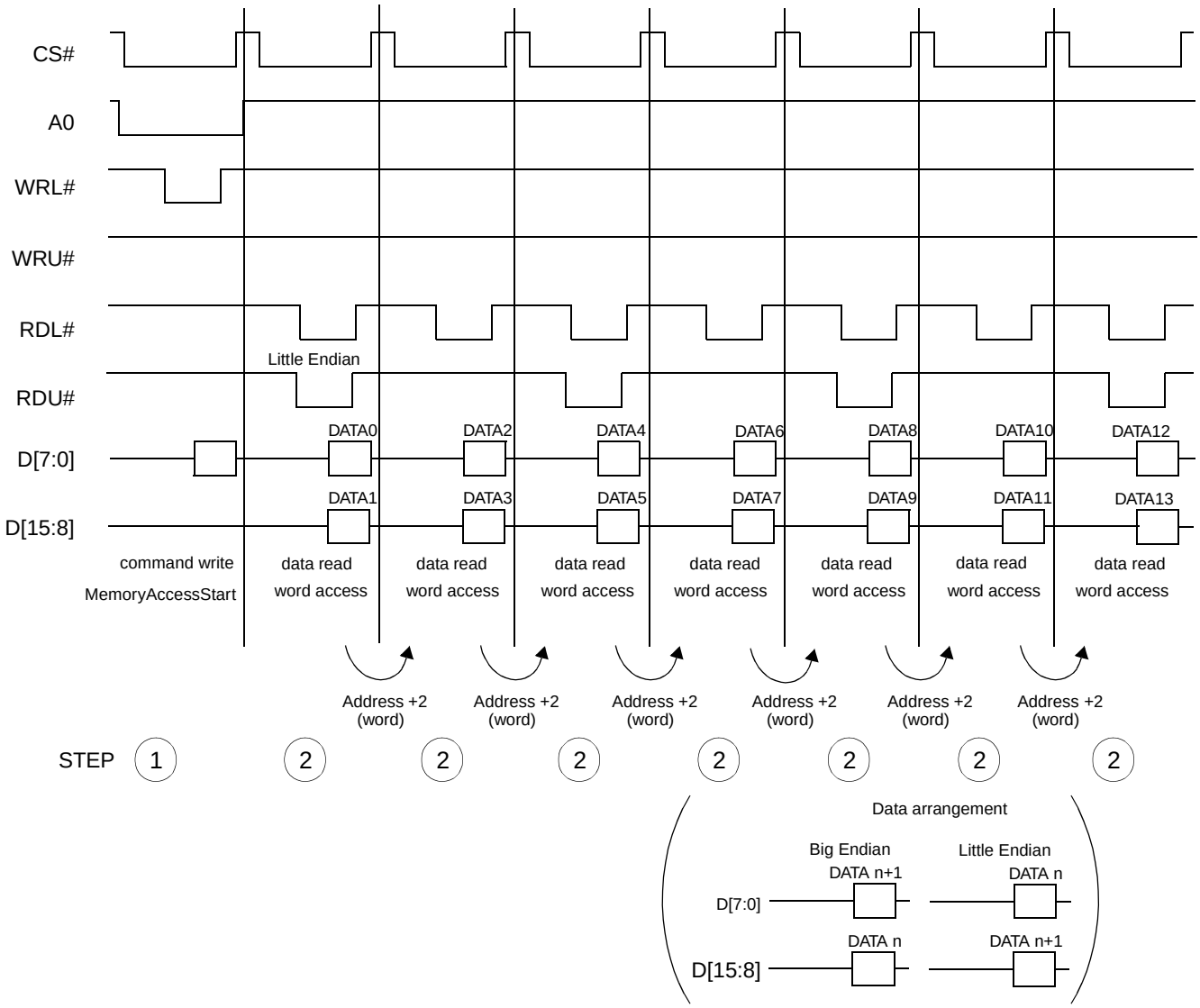


Figure 15-12 Sample timing of “memory read” for Mode 80 when Memory Access Select Enabled

1. write register address of Memory Access Start register (REG[C4h] (command write).

Note

No “data write” is required after a command write to the memory Access Start register (REG[C4h]). This step configures the S1D13708 for burst memory access beginning with the next data read.

2. read Memory data (data write)
   If the Memory Access Select bit (REG[C6h] bit 0 = 1), memory accesses are word accesses even if RDU# is high (RDU# is ignored and RDL# is used to write both the upper and lower bytes). The bit/Little endian setting is used to determine the data

arrangement for word accesses only.

The S1D13708 indirect interface implements an auto increment function to allow burst memory accesses. For byte accesses, the Memory Address Pointer registers (REG[C0h], REG[C1h], REG[C2h]) are automatically incremented “+1”. For word accesses, the Memory Address Pointer registers are automatically incremented “+2”.

Note

If the Memory Access Select bit is enabled (REG[C6h] bit 0 = 1), all memory accesses are word accesses (RDU# is ignored). Therefore, the memory address set in REG[C0h] through REG[C2h] must be an even address.

15.3 Limitations

Each Indirect cycle requires a certain number of BCLK cycles to setup/complete (refer to Indirect Interface Timing (Mode 68) on page 62 and Indirect Interface Timing (Mode 80) on page 64). The BCLK source can be derived from either XTAL or CLKI. When XTAL is used, and has a different frequency than the CPU, the CPU must take the speed difference into account and change the cycle time accordingly.

Example: If the CPU clock is running at 20MHz and the S1D13708 is running with a 10MHz crystal, referring to Table 6-14, item t6a requires 7.5 BUSCLKs. Since the CPU will be running 2 times faster than BUSCLK, t6a will require to be doubled to15 BUSCLKs.

16 Embedded Crystal Oscillator

The S1D13708 includes an embedded crystal oscillator which is available when the Indirect Interface is selected.

16.1 Oscillator Circuit

When a crystal is used in conjunction with the S1D13708, the designer must connect an external resistor and capacitors as recommended in the following figure.

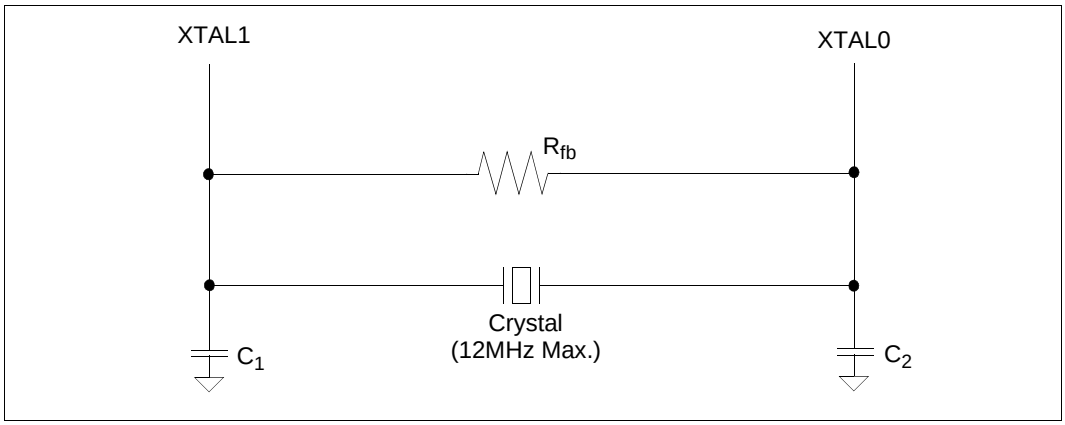


Figure 16-1 Recommended Crystal Oscillator Circuit

Where R_{fb} is a feedback resistor, and C_1 and C_2 are load capacitances. To determine the appropriate values for the system, a good starting point is 1 megOhm (MΩ) for R_{fb}, 10 picoFarads (pF) for C_1, and 10 picoFarads (pF) for C_2. The values of R_{fb}, C_1, and C_2 should be further refined to meet the frequency requirements of the system.

Note

The XTAL Enable bits (REG[CAh] bits 5 and 4) are used to enable the oscillator circuit. See the BCLK Source Select Register on page 157 for further information on these two bits.

17 Big-Endian Bus Interface

17.1 Byte Swapping Bus Data

The display buffer and register architecture of the S1D13708 is inherently little-endian. If configured as big-endian (CNF4 = 1 at reset), bus accesses are automatically handled by byte swapping all read/write data to/from the internal display buffer and registers.

Bus data byte swapping translates all byte accesses correctly to the S1D13708 register and display buffer locations. To maintain the correct translation for 16-bit word access, even address bytes must be mapped to the MSB of the 16-bit word, and odd address bytes to the LSB of the 16-bit word. For example:

Byte write 11h to register address 1Eh -> REG[1Eh] <= 11h
Byte write 22h to register address 1Fh -> REG[1Fh] <= 22h

Word write 1122h to register address 1Eh-> REG[1Eh] <= 11h
REG[1Fh] <= 22h

17.1.1 16 Bpp Color Depth

For 16 bpp color depth, the Display Data Byte Swap bit (REG[71h] bit 6) must be set to 1.

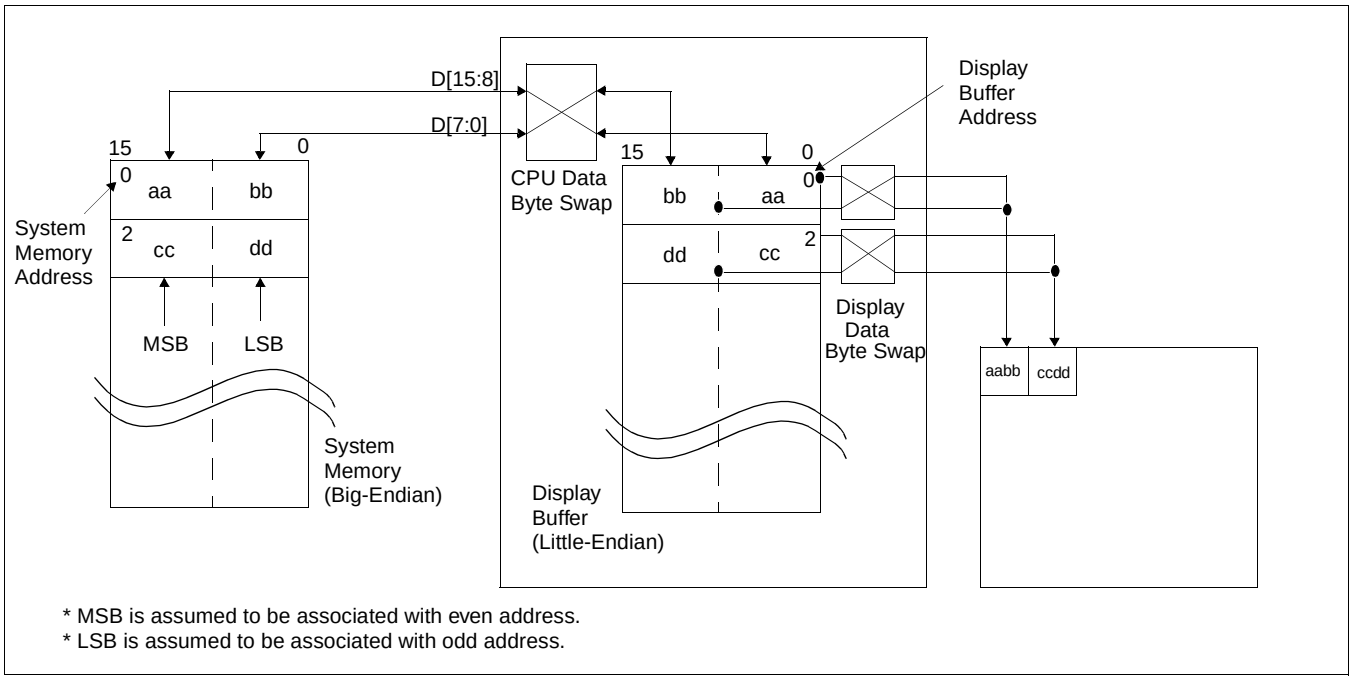


Figure 17-1 Byte-swapping for 16 Bpp

For 16 bpp color depth, the MSB of the 16-bit pixel data is stored at the even system memory address location and the LSB of the 16-bit pixel data is stored at the odd system memory address location. Bus data byte swapping (automatic when the S1D13708 is configured for Big-Endian) causes the 16-bit pixel data to be stored byte-swapped in the S1D13708 display buffer. During display refresh this stored data must be byte-swapped again before it is sent to the display.

17.1.2 1/2/4/8 Bpp Color Depth

For 1/2/4/8 bpp color depth, byte swapping must be performed on the bus data but not the display data.

For 1/2/4/8 bpp color depth, the Display Data Byte Swap bit (REG[71h] bit 6) must be set to 0.

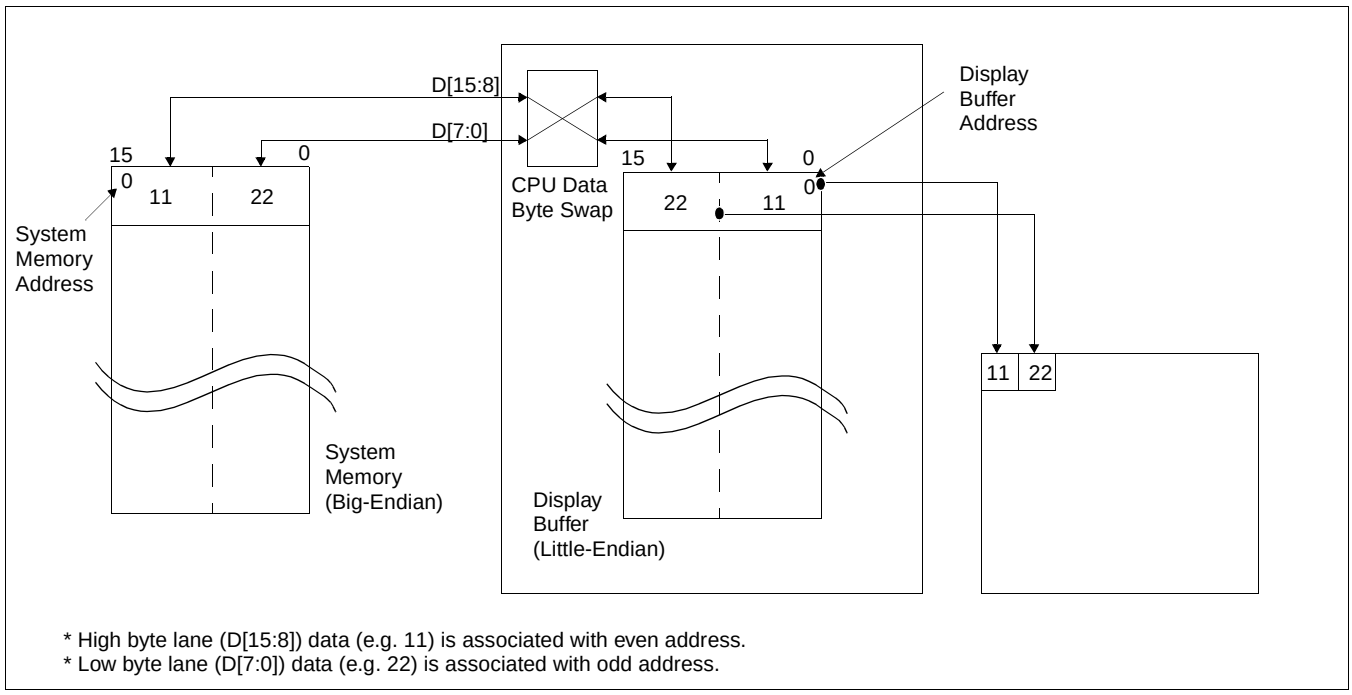


Figure 17-2 Byte-swapping for 1/2/4/8 Bpp

18 Power Save Mode

A software initiated Power Save Mode is incorporated into the S1D13708 to accommodate the need for power reduction in the hand-held devices market. This mode is enabled via the Power Save Mode Enable bit (REG[A0h] bit 0).

Software Power Save Mode saves power by powering down the panel and stopping display refresh accesses to the display buffer.

Table 18-1: Power Save Mode Function Summary

	Software Power Save	Normal
IO Access Possible?	Yes	Yes
Memory Writes Possible?	Yes[1]	Yes
Memory Reads Possible?	No[1]	Yes
Look-Up Table Registers Access Possible?	Yes	Yes
Display Active?	No	Yes
LCD Interface Outputs	Forced Low	Active
PWMCLK	Stopped	Active
GPIO Pins configured for HR-TFT/D-TFD/TFT Type 2/3[2]	Forced Low	Active
GPIO Pins configured as GPIOs Access Possible?	Yes[3]	Yes

Note

[1] When power save mode is enabled, the memory controller is powered down and the status of the memory controller is indicated by the Memory Controller Power Save Status bit (REG[A0h] bit 3). However, memory writes are possible during power save mode because the S1D13708 dynamically enables the memory controller for display buffer writes.

[2] GPIO Pins are configured using the configuration pin CNF3 which is latched on the rising edge of RESET#. For information on CNF3, see Table 4-8: “Summary of Power-On/Reset Options,” on page 38.

[3] GPIOs can be accessed and if configured as outputs can be changed.

After reset, the S1D13708 is always in Power Save Mode. Software must initialize the chip (i.e. programs all registers) and then clear the Power Save Mode Enable bit.

19 Mechanical Data

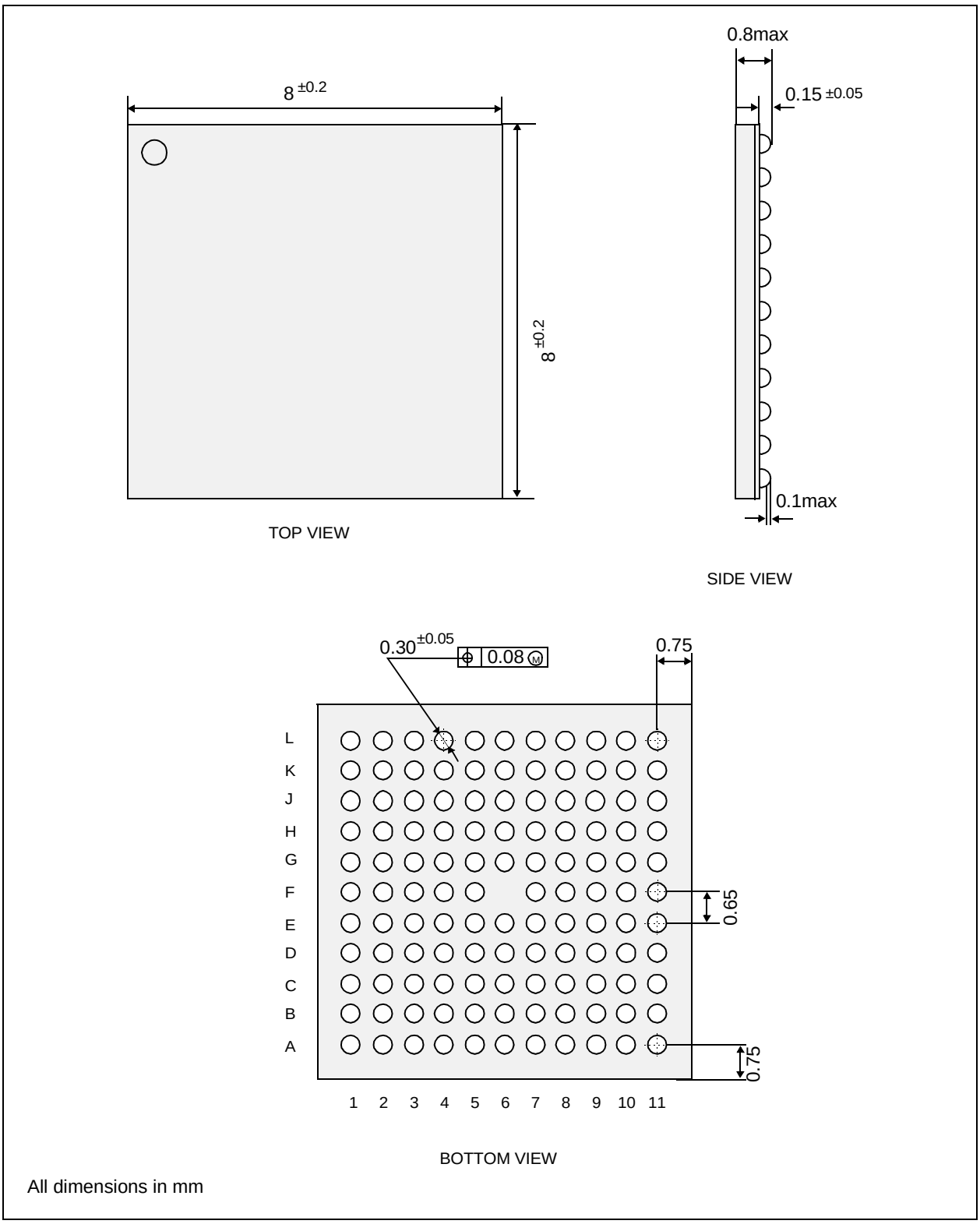


Figure 19-1 Mechanical Data PFBGA 120-pin Package

20 References

The following is a partial list of documents which contain additional information related to the S1D13708. Document numbers are listed in parenthesis after the document name. All documentation for the S1D13708 can be found at the Epson Research and Development Website at **www.erd.epson.com**.

- 13708CFG Configuration Utility Users Manual (X39A-B-001-xx)
- 13708PLAY Diagnostic Utility Users Manual (X39A-B-002-xx)
- 13708BMP Demonstration Program Users Manual (X39A-B-003-xx)
- S1D13708 Product Brief (X39A-C-001-xx)
- S1D13708 Programming Notes And Examples (X39A-G-003-xx)
- S1D13708 Power Consumption (X39A-G-006-xx)
- Interfacing to the NEC VR4102/VR4111 Microprocessors (X39A-G-007-xx)
- Interfacing to the NEC VR4181 Microprocessor (X39A-G-008-xx)
- Interfacing to the Motorola MPC821 Microprocessor (X39A-G-009-xx)
- Interfacing to the Motorola MCF5307 “Coldfire” Microprocessors (X39A-G-010-xx)
- Connecting to the Sharp HR-TFT Panels (X39A-G-011-xx)
- Interfacing to the Motorola RedCap2 DSP (X39A-G-014-xx)
- Interfacing to 8-Bit Processors (X39A-G-015-xx)
- Interfacing to the Motorola MC68VZ328 Dragonball Microprocessor (X39A-G-016-xx)
- Interfacing to the Intel StrongARM SA-1110 Microprocessor (X39A-G-019-xx)
- S1D13708 Register Summary (X39A-R-001-xx)

21 Technical Support

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp/

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com/

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de/

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

Programming Notes and Examples

Document Number: X39A-G-003-01

THIS PAGE LEFT BLANK

Table of Contents

List of Tables

THIS PAGE LEFT BLANK

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This guide discusses programming issues and provides examples for the main features of the S1D13708, such as SwivelView and Picture-in-Picture Plus (PiP$^+$). The example source code referenced in this guide is available on the web at www.erd.epson.com.

This guide also introduces the Hardware Abstraction Layer (HAL), which is designed to simplify the programming of the S1D13708. Most S1D13xxx products have HAL support, thus allowing OEMs to do multiple designs with a common code base.

This document is updated as appropriate. Please check the Epson Research and Development website at www.erd.epson.com for the latest revision of this document and source before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Identifying the S1D13708

The S1D13708 can be identified by reading the value contained in the Revision Code Register (REG[00h]). To identify the S1D13708 follow the steps below.

1. Read REG[00h].
2. The production version of the S1D13708 returns a value of 34h. This value can be broken down into the following.
   a. The product code for the S1D13708 is 0Dh (001101 binary) and can be found in bits 7-2.
   b. The revision code is 0h (00 binary) and can be found in bits 1-0.

3 Initialization

This section describes how to initialize the S1D13708. Sample code for performing initialization of the S1D13708 is provided in the file **init13708.c** which is available on the internet at www.erd.epson.com.

S1D13708 initialization can be broken into the following steps.

1. Set all registers to initial values. The values are obtained by using the s1d13708.h file that is exported by the **13708CFG.EXE** configuration utility. For more information on 13708CFG, see the *13708CFG User Manual*, document number X39A-B-001-xx.

2. Program the Look-Up Table (LUT) with color values. For details on programming the LUT, see Section 5, “Look-Up Table (LUT)” on page 15.

3. Clear the display buffer.

If the system implementation uses a clock chip instead of a fixed oscillator, refer to the HAL (Hardware Abstraction Layer) sample code available on the internet at www.erd.epson.com. The Epson S5U13708B00B evaluation board uses a Cypress clock chip.

4 Memory Models

The S1D13708 contains a display buffer of 80K bytes and supports color depths of 1, 2, 4, 8, and 16 bit-per-pixel. For each color depth, the data format is packed pixel.

Packed pixel data may be envisioned as a stream of pixels. In this stream, pixels are packed adjacent to each other. If a pixel requires four bits, then it is located in the four most significant bits of a byte. The pixel to the immediate right on the display occupies the lower four bits of the same byte. The next two pixels to the immediate right are located in the following byte, etc.

4.1 Memory Organization for One Bit-per-pixel (2 Colors/Gray Shades)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Pixel 0	Pixel 1	Pixel 2	Pixel 3	Pixel 4	Pixel 5	Pixel 6	Pixel 7

Figure 4-1: Pixel Storage for 1 Bpp in One Byte of Display Buffer

At a color depth of 1 bpp, each byte of display buffer contains eight adjacent pixels. Setting or resetting any pixel requires reading the entire byte, masking out the unchanged bits and setting the appropriate bits to 1.

One bit pixels provide 2 gray shades/color possibilities. For monochrome panels the gray shades are generated by indexing into the first two elements of the green component of the Look-Up Table (LUT). For color panels the 2 colors are derived by indexing into the first 2 positions of the LUT.

4.2 Memory Organization for Two Bit-per-pixel (4 Colors/Gray Shades)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Pixel 0 Bits 1-0		Pixel 1 Bits 1-0		Pixel 2 Bits 1-0		Pixel 3 Bits 1-0	

Figure 4-2: Pixel Storage for 2 Bpp in One Byte of Display Buffer

At a color depth of 2 bpp, each byte of display buffer contains four adjacent pixels. Setting or resetting any pixel requires reading the entire byte, masking out the unchanged bits and setting the appropriate bits to 1.

Two bit pixels provide 4 gray shades/color possibilities. For monochrome panels the gray shades are generated by indexing into the first 4 elements of the green component of the Look-Up Table (LUT). For color panels the 4 colors are derived by indexing into the first 4 positions of the LUT.

4.3 Memory Organization for Four Bit-per-pixel (16 Colors/Gray Shades)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Pixel 0 Bits 3-0				Pixel 1 Bits 3-0			

Figure 4-3: Pixel Storage for 4 Bpp in One Byte of Display Buffer

At a color depth of 4 bpp, each byte of display buffer contains two adjacent pixels. Setting or resetting any pixel requires reading the entire byte, masking out the upper or lower nibble (4 bits) and setting the appropriate bits to 1.

Four bit pixels provide 16 gray shades/color possibilities. For monochrome panels the gray shades are generated by indexing into the first 16 elements of the green component of the Look-Up Table (LUT). For color panels the 16 colors are derived by indexing into the first 16 positions of the LUT.

4.4 Memory Organization for 8 Bpp (256 Colors/64 Gray Shades)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Pixel 0 Bits 7-0							

Figure 4-4: Pixel Storage for 8 Bpp in One Byte of Display Buffer

At a color depth of 8 bpp, each byte of display buffer represents one pixel on the display. At this color depth the read-modify-write cycles are eliminated making pixel updates faster.

Each byte indexes into one of the 256 positions of the LUT. The S1D13708 LUT supports six bits per primary color. This translates into 256K possible colors when color mode is selected. Therefore the display has 256 colors available out of a possible 256K colors.

When a monochrome panel is selected, the green component of the LUT is used to determine the intensity. The green LUT, with six bits, can resolve 64 unique gray shades of a possible 256. It is recommended that LUT indexes 00h through 3Fh be used for the 64 gray shades to be displayed.

4.5 Memory Organization for 16 Bpp (65536 Colors/64 Gray Shades)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Red Component Bits 4-0					Green Component Bits 5-3		
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Green Component Bits 2-0			Blue Component Bits 4-0				

Figure 4-5: Pixel Storage for 16 Bpp in Two Bytes of Display Buffer

At a color depth of 16 bpp the S1D13708 is capable of displaying 64K (65536) colors. The 64K color pixel is divided into three parts: five bits for red, six bits for green, and five bits for blue. In this mode the LUT is bypassed and output goes directly into the Frame Rate Modulator.

Should monochrome mode be chosen at this color depth, the output sends the six bits of the green component to the modulator for a total of 64 possible gray shades. Note that since 8 bpp also provides 64 gray shades, it is recommended to use 8 bpp for monochrome mode in order to reduce the amount of display memory used.

5 Look-Up Table (LUT)

This section discusses programming the S1D13708 Look-Up Table (LUT). Included is a summary of the LUT registers, recommendations for color/gray shade LUT values, and additional programming considerations. For a discussion of the LUT architecture, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

The S1D13708 is designed with a LUT consisting of 256 indexed red/green/blue entries. Each LUT entry is six bits wide. The color depth (bpp) determines how many indices are used. For example, 1 bpp uses the first 2 indices, 2 bpp uses the first 4 indices, 4 bpp uses the first 16 indices and 8 bpp uses all 256 indices. 16 bpp bypasses the LUT.

In color modes, the pixel values stored in the display buffer index directly to an RGB value stored in the LUT. In monochrome modes, the pixel value indexes into the green component of the LUT and the amount of green at that index controls the intensity.

5.1 Registers

5.1.1 Look-Up Table Registers

Look-Up Table Blue Write Data Register
REG[08h] Write Only

LUT Blue Write Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Blue Write Data Bits [5:0]
This register contains the data to be written to the blue component of the Look-Up Table. The data is stored in this register until a write to the LUT Write Address register (REG[0Bh]) moves the data into the Look-Up Table.

Note
The LUT entry is updated only when the LUT Write Address Register (REG[0Bh]) is written to.

Look-Up Table Green Write Data Register
REG[09h] Write Only

LUT Green Write Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Green Write Data Bits [5:0]
This register contains the data to be written to the green component of the Look-Up Table. The data is stored in this register until a write to the LUT Write Address register (REG[0Bh]) moves the data into the Look-Up Table.

Note

The LUT entry is updated only when the LUT Write Address Register (REG[0Bh]) is written to.

Look-Up Table Red Write Data Register
REG[0Ah] Write Only

LUT Red Write Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Red Write Data Bits [5:0]

This register contains the data to be written to the red component of the Look-Up Table. The data is stored in this register until a write to the LUT Write Address register (REG[0Bh]) moves the data into the Look-Up Table.

Note

The LUT entry is updated only when the LUT Write Address Register (REG[0Bh]) is written to.

Look-Up Table Write Address Register
REG[0Bh] Write Only

LUT Write Address Bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 LUT Write Address Bits [7:0]

This register forms a pointer into the Look-Up Table (LUT) which is used to write LUT data stored in REG[08h], REG[09h], and REG[0Ah]. **The data is updated to the LUT only with the completion of a write to this register**. This is a write-only register and returns 00h if read.

Look-Up Table Blue Read Data Register
REG[0Ch] Read Only

LUT Blue Read Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Blue Read Data Bits [5:0]

This register contains the data from the blue component of the Look-Up Table. The LUT entry read is controlled by the LUT Read Address Register (REG[0Fh]). This is a read-only register.

Note

This register is updated only when the LUT Read Address Register (REG[0Fh]) is written to.

Look-Up Table Green Read Data Register
REG[0Dh] Read Only

LUT Green Read Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Green Read Data Bits [5:0]
This register contains the data from the green component of the Look-Up Table. The LUT entry read is controlled by the LUT Read Address Register (REG[0Fh]). This is a read-only register.

Note
This register is updated only when the LUT Read Address Register (REG[0Fh]) is written to.

Look-Up Table Red Read Data Register
REG[0Eh] Read Only

LUT Red Read Data Bits 5-0						n/a	
7	6	5	4	3	2	1	0

bits 7-2 LUT Red Read Data Bits [5:0]
This register contains the data from the red component of the Look-Up Table. The LUT entry read is controlled by the LUT Read Address Register (REG[0Fh]). This is a read-only register.

Note
This register is updated only when the LUT Read Address Register (REG[0Fh]) is written to.

Look-Up Table Read Address Register
REG[0Fh] Write Only

LUT Read Address Bits 7-0							
7	6	5	4	3	2	1	0

bits 7-0 LUT Read Address Bits [7:0]
This register forms a pointer into the Look-Up Table (LUT) which is used to read LUT data and store it in REG[0Ch], REG[0Dh], REG[0Eh]. **The data is read from the LUT only when a write to this register is completed**. This is a write-only register and returns 00h if read.

5.2 Look-Up Table Organization

- The Look-Up Table treats the value of a pixel as an index into an array. For example, a pixel value of zero would point to the first LUT entry, whereas a pixel value of seven would point to the eighth LUT entry.
- The value contained in each LUT entry represents the intensity of the given color or gray shade.
- The S1D13708 Look-Up Table is linear. This means increasing the LUT entry number results in a brighter color or gray shade. For example, a LUT entry of FCh in the red bank results in bright red output while a LUT entry of 1Ch results in dull red.

Table 5-1: Look-Up Table Configurations

Color Depth	Look-Up Table Indices Used			Effective Gray Shades/Colors
	RED	GREEN	BLUE	
1 bpp gray		2		2 gray shades
2 bpp gray		4		4 gray shades
4 bpp gray		16		16 gray shades
8 bpp gray		256		64 gray shades
16 bpp gray				64 gray shades
1 bpp color	2	2	2	2 colors
2 bpp color	4	4	4	4 colors
4 bpp color	16	16	16	16 colors
8 bpp color	256	256	256	256 colors
16 bpp color				65536 colors

= Indicates the Look-Up Table is not used for that display mode

5.2.1 Gray Shade Modes

Gray shade (monochrome) modes are defined by the Color/Mono Panel Select bit (REG[10h] bit 6). When this bit is set to 0, the value output to the panel is derived solely from the green component of the pixel.

For each bits-per-pixel mode (excluding 16 bpp) a table of sample LUT values is provided. These LUT values are a standardized set of intensities used by the Epson S1D13708 utility programs.

Note
These LUT values carry eight bits of significance. The S1D13708 LUT uses only the six most-significant bits. The 2 least-significant bits are ignored.

1 bpp gray shade

The 1 bpp gray shade mode uses the green component of the first 2 LUT entries. The remaining indices of the LUT are unused.

Table 5-2: Suggested LUT Values for 1 Bpp Gray Shade

Index	Red	Green	Blue
00	00	00	00
01	00	FF	00
02	00	00	00
...	00	00	00
FF	00	00	00

Unused entries

2 bpp gray shade

The 2 bpp gray shade mode uses the green component of the first 4 LUT entries. The remaining indices of the LUT are unused.

Table 5-3: Suggested LUT Values for 4 Bpp Gray Shade

Index	Red	Green	Blue
00	00	00	00
01	00	55	00
02	00	AA	00
03	00	FF	00
04	00	00	00
...	00	00	00
FF	00	00	00

Unused entries

4 bpp gray shade

The 4 bpp gray shade mode uses the green component of the first 16 LUT entries. The remaining indices of the LUT are unused.

Table 5-4: Suggested LUT Values for 4 Bpp Gray Shade

Index	Red	Green	Blue
00	00	00	00
01	00	11	00
02	00	22	00
03	00	33	00
04	00	44	00
05	00	55	00
06	00	66	00
07	00	77	00
08	00	88	00
09	00	99	00
0A	00	AA	00
0B	00	BB	00
0C	00	CC	00
0D	00	DD	00
0E	00	EE	00
0F	00	FF	00
10	00	00	00
...	00	00	00
FF	00	00	00

Unused entries

8 bpp gray shade

When configured for 8 bpp gray shade mode, the green component of all 256 LUT entries may be used. However, the green component alone only provides 64 intensities (6 bits).

Table 5-5: Suggested LUT Values for 8 Bpp Gray Shade

Index	Red	Green	Blue	Index	Red	Green	Blue
00	00	00	00	20	00	80	00
01	00	04	00	21	00	84	00
02	00	08	00	22	00	88	00
03	00	0C	00	23	00	8C	00
04	00	10	00	24	00	90	00
05	00	14	00	25	00	94	00
06	00	18	00	26	00	98	00
07	00	1C	00	27	00	9C	00
08	00	20	00	28	00	A0	00
09	00	24	00	29	00	A4	00
0A	00	28	00	2A	00	A8	00
0B	00	2C	00	2B	00	AC	00
0C	00	30	00	2C	00	B0	00
0D	00	34	00	2D	00	B4	00
0E	00	38	00	2E	00	B8	00
0F	00	3C	00	2F	00	BC	00
10	00	40	00	30	00	C0	00
11	00	44	00	31	00	C4	00
12	00	48	00	32	00	C8	00
13	00	4C	00	33	00	CC	00
14	00	50	00	34	00	D0	00
15	00	54	00	35	00	D4	00
16	00	58	00	36	00	D8	00
17	00	5C	00	37	00	DC	00
18	00	60	00	38	00	E0	00
19	00	64	00	39	00	E4	00
1A	00	68	00	3A	00	E8	00
1B	00	6C	00	3B	00	EC	00
1C	00	70	00	3C	00	F0	00
1D	00	74	00	3D	00	F4	00
1E	00	78	00	3E	00	F8	00
1F	00	7C	00	3F	00	FC	00
				40	00	00	00
				...	00	00	00
				FF	00	00	00

Unused entries

16 bpp gray shade

The Look-Up Table is bypassed at this color depth, therefore programming the LUT is not required.

As with 8 bpp there are limitations to the colors which can be displayed. In this mode the six bits of green are used to set the absolute intensity of the image. This results in 64 gray shades.

5.2.2 Color Modes

In color display modes, the number of LUT entries used is determined by the color depth.

For each color depth (excluding 16 bpp) a table of sample LUT values is provided. These LUT values are a standardized set of colors used by the Epson S1D13708 utility programs.

Note
The S1D13708 LUT uses only the six most-significant bits of the LUT. The 2 least-significant bits are ignored.

1 bpp color

When the S1D13708 is configured for 1 bpp color mode the first 2 entries in the LUT are used. The remaining indices of the LUT are unused.

Table 5-6: Suggested LUT Values for 1 bpp Color

Index	Red	Green	Blue
00	00	00	00
01	FF	FF	FF
02	00	00	00
...	00	00	00
FF	00	00	00

= Indicates unused entries in the LUT

2 bpp color

When the S1D13708 is configured for 2 bpp color mode the first 4 entries in the LUT are used. The remaining indices of the LUT are unused.

Table 5-7: Suggested LUT Values for 2 bpp Color

Index	Red	Green	Blue
00	00	00	00
01	00	00	FF
02	FF	00	00
03	FF	FF	FF
04	00	00	00
...	00	00	00
FF	00	00	00

= Indicates unused entries in the LUT

4 bpp color

When the S1D13708 is configured for 4 bpp color mode the first 16 entries in the LUT are used. The remaining indices of the LUT are unused.

The following table shows LUT values that simulate those of a VGA operating in 16 color mode.

Table 5-8: Suggested LUT Values for 4 bpp Color

Index	Red	Green	Blue
00	00	00	00
01	00	00	AA
02	00	AA	00
03	00	AA	AA
04	AA	00	00
05	AA	00	AA
06	AA	AA	00
07	AA	AA	AA
08	00	00	00
09	00	00	FF
0A	00	FF	00
0B	00	FF	FF
0C	FF	00	00
0D	FF	00	FF
0E	FF	FF	00
0F	FF	FF	FF
10	00	00	00
...	00	00	00
FF	00	00	00

= Indicates unused entries in the LUT

8 bpp color

When the S1D13708 is configured for 8 bpp color mode all 256 entries in the LUT are used.

The S1D13708 LUT has six bits (64 values) of intensity control per primary color which is the same as a standard VGA RAMDAC.

The following table shows LUT values that simulate the VGA default color palette.

Table 5-9: Suggested LUT Values for 8 bpp Color

Index	R	G	B	Index	R	G	B	Index	R	G	B	Index	R	G	B
00	00	00	00	40	00	00	00	80	FF	FF	00	C0	00	00	00
01	00	00	AA	41	00	00	11	81	FF	EF	00	C1	00	11	11
02	00	AA	00	42	00	00	22	82	FF	DE	00	C2	00	22	22
03	00	AA	AA	43	00	00	33	83	FF	CD	00	C3	00	33	33
04	AA	00	00	44	00	00	44	84	FF	BC	00	C4	00	44	44
05	AA	00	AA	45	00	00	55	85	FF	AB	00	C5	00	55	55
06	AA	AA	00	46	00	00	66	86	FF	9A	00	C6	00	66	66
07	AA	AA	AA	47	00	00	77	87	FF	89	00	C7	00	77	77
08	55	55	55	48	00	00	89	88	FF	77	00	C8	00	89	89
09	00	00	FF	49	00	00	9A	89	FF	66	00	C9	00	9A	9A
0A	00	FF	00	4A	00	00	AB	8A	FF	55	00	CA	00	AB	AB
0B	00	FF	FF	4B	00	00	BC	8B	FF	44	00	CB	00	BC	BC
0C	FF	00	00	4C	00	00	CD	8C	FF	33	00	CC	00	CD	CD
0D	FF	00	FF	4D	00	00	DE	8D	FF	22	00	CD	00	DE	DE
0E	FF	FF	00	4E	00	00	EF	8E	FF	11	00	CE	00	EF	EF
0F	FF	FF	FF	4F	00	00	FF	8F	FF	00	00	CF	00	FF	FF
10	00	00	00	50	00	00	FF	90	FF	00	00	D0	FF	00	00
11	11	11	11	51	00	11	FF	91	FF	00	11	D1	FF	11	11
12	22	22	22	52	00	22	FF	92	FF	00	22	D2	FF	22	22
13	33	33	33	53	00	33	FF	93	FF	00	33	D3	FF	33	33
14	44	44	44	54	00	44	FF	94	FF	00	44	D4	FF	44	44
15	55	55	55	55	00	55	FF	95	FF	00	55	D5	FF	55	55
16	66	66	66	56	00	66	FF	96	FF	00	66	D6	FF	66	66
17	77	77	77	57	00	77	FF	97	FF	00	77	D7	FF	77	77
18	89	89	89	58	00	89	FF	98	FF	00	89	D8	FF	89	89
19	9A	9A	9A	59	00	9A	FF	99	FF	00	9A	D9	FF	9A	9A
1A	AB	AB	AB	5A	00	AB	FF	9A	FF	00	AB	DA	FF	AB	AB
1B	BC	BC	BC	5B	00	BC	FF	9B	FF	00	BC	DB	FF	BC	BC
1C	CD	CD	CD	5C	00	CD	FF	9C	FF	00	CD	DC	FF	CD	CD
1D	DE	DE	DE	5D	00	DE	FF	9D	FF	00	DE	DD	FF	DE	DE
1E	EF	EF	EF	5E	00	EF	FF	9E	FF	00	EF	DE	FF	EF	EF
1F	FF	FF	FF	5F	00	FF	FF	9F	FF	00	FF	DF	FF	FF	FF
20	00	00	00	60	00	FF	FF	A0	FF	00	FF	E0	00	FF	00
21	11	00	00	61	00	FF	EF	A1	EF	00	FF	E1	11	FF	11
22	22	00	00	62	00	FF	DE	A2	DE	00	FF	E2	22	FF	22
23	33	00	00	63	00	FF	CD	A3	CD	00	FF	E3	33	FF	33
24	44	00	00	64	00	FF	BC	A4	BC	00	FF	E4	44	FF	44

Table 5-9: Suggested LUT Values for 8 bpp Color (Continued)

Index	R	G	B	Index	R	G	B	Index	R	G	B	Index	R	G	B
25	55	00	00	65	00	FF	AB	A5	AB	00	FF	E5	55	FF	55
26	66	00	00	66	00	FF	9A	A6	9A	00	FF	E6	66	FF	66
27	77	00	00	67	00	FF	89	A7	89	00	FF	E7	77	FF	77
28	89	00	00	68	00	FF	77	A8	77	00	FF	E8	89	FF	89
29	9A	00	00	69	00	FF	66	A9	66	00	FF	E9	9A	FF	9A
2A	AB	00	00	6A	00	FF	55	AA	55	00	FF	EA	AB	FF	AB
2B	BC	00	00	6B	00	FF	44	AB	44	00	FF	EB	BC	FF	BC
2C	CD	00	00	6C	00	FF	33	AC	33	00	FF	EC	CD	FF	CD
2D	DE	00	00	6D	00	FF	22	AD	22	00	FF	ED	DE	FF	DE
2E	EF	00	00	6E	00	FF	11	AE	11	00	FF	EE	EF	FF	EF
2F	FF	00	00	6F	00	FF	00	AF	00	00	FF	EF	FF	FF	FF
30	00	00	00	70	00	FF	00	B0	00	00	00	F0	00	00	FF
31	00	11	00	71	11	FF	00	B1	11	00	11	F1	11	11	FF
32	00	22	00	72	22	FF	00	B2	22	00	22	F2	22	22	FF
33	00	33	00	73	33	FF	00	B3	33	00	33	F3	33	33	FF
34	00	44	00	74	44	FF	00	B4	44	00	44	F4	44	44	FF
35	00	55	00	75	55	FF	00	B5	55	00	55	F5	55	55	FF
36	00	66	00	76	66	FF	00	B6	66	00	66	F6	66	66	FF
37	00	77	00	77	77	FF	00	B7	77	00	77	F7	77	77	FF
38	00	89	00	78	89	FF	00	B8	89	00	89	F8	89	89	FF
39	00	9A	00	79	9A	FF	00	B9	9A	00	9A	F9	9A	9A	FF
3A	00	AB	00	7A	AB	FF	00	BA	AB	00	AB	FA	AB	AB	FF
3B	00	BC	00	7B	BC	FF	00	BB	BC	00	BC	FB	BC	BC	FF
3C	00	CD	00	7C	CD	FF	00	BC	CD	00	CD	FC	CD	CD	FF
3D	00	DE	00	7D	DE	FF	00	BD	DE	00	DE	FD	DE	DE	FF
3E	00	EF	00	7E	EF	FF	00	BE	EF	00	EF	FE	EF	EF	FF
3F	00	FF	00	7F	FF	FF	00	BF	FF	00	FF	FF	FF	FF	FF

16 bpp color

The Look-Up Table is bypassed at this color depth, therefore programming the LUT is not required.

6 Power Save Mode

The S1D13708 is designed for very low-power applications. During normal operation, the internal clocks are dynamically disabled when not required. The S1D13708 design also includes a Power Save Mode to further save power. When Power Save Mode is initiated, LCD power sequencing is required to ensure the LCD bias power supply is disabled properly. For further information on LCD power sequencing, see Section 6.3, "LCD Power Sequencing" on page 29.

For Power Save Mode AC Timing, see the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

6.1 Overview

The S1D13708 includes a software initiated Power Save Mode. Enabling/disabling Power Save Mode is controlled using the Power Save Mode Enable bit (REG[A0h] bit 0).

While Power Save Mode is enabled the following conditions apply.

- Registers are accessible
- Memory writes are possible[1]
- Memory reads are not possible
- Look-Up Table registers are accessible.
- LCD display is inactive.
- LCD interface outputs are forced low.

Note

[1] Memory writes are possible during power save mode because the S1D13708 dynamically enables the memory controller for display buffer writes.

6.2 Registers

6.2.1 Power Save Mode Enable

Power Save Configuration Register REG[A0h]							Read/Write
Vertical Non-Display Period Status (RO)	n/a			Memory Controller Power Save Status (RO)	n/a		Power Save Mode Enable
7	6	5	4	3	2	1	0

The Power Save Mode Enable bit initiates Power Save Mode when set to 1. Setting the bit to 0 disables Power Save Mode and returns the S1D13708 to normal mode. At reset this bit is set to 1.

Note

Enabling/disabling Power Save Mode requires proper LCD Power Sequencing. See Section 6.3, “LCD Power Sequencing” on page 29.

6.2.2 Memory Controller Power Save Status

Power Save Configuration Register REG[A0h]							Read/Write
Vertical Non-Display Period Status (RO)	n/a			Memory Controller Power Save Status (RO)	n/a		Power Save Mode Enable
7	6	5	4	3	2	1	0

The Memory Controller Power Save Status bit is a read-only status bit which indicates the power save state of the S1D13708 SRAM interface. When this bit returns a 1, the SRAM interface is powered down and the memory clock source may be disabled. When this bit returns a 0, the SRAM interface is active. This bit returns a 0 after a chip reset.

Note

Memory writes are possible during power save mode because the S1D13708 dynamically enables the memory controller for display buffer writes.

6.3 LCD Power Sequencing

The S1D13708 requires LCD power sequencing (the process of powering-on and powering-off the LCD panel). LCD power sequencing allows the LCD bias voltage to discharge prior to shutting down the LCD signals, preventing long term damage to the panel and avoiding unsightly "lines" at power-on/power-off.

Proper LCD power sequencing for power-off requires a delay from the time the LCD power is disabled to the time the LCD signals are shut down. Power-on requires the LCD signals to be active prior to applying power to the LCD. This time interval depends on the LCD bias power supply design. For example, the LCD bias power supply on the S5U13708B00B Evaluation Board requires about 50 ms to fully discharge. Other power supply designs may vary.

This section assumes the LCD bias power is controlled through GPO0. The S1D13708 GPIO pins are multi-use pins and may not be available in all system designs. For further information on the availability of GPIO pins, see the *S1D13708 Hardware Functional Specification*, document number X36A-A-001-xx.

Note

This section discusses LCD power sequencing for passive and TFT (non-HR-TFT) panels only.

For further information on LCD power sequencing the HR-TFT, see *Connecting to the Sharp HR-TFT Panels*, document number X36A-G-011-xx.

For further information on LCD Power Sequencing the D-TFD, see *Connecting to the Epson D-TFD Panels*, document number X36A-G-012-xx.

6.4 Enabling Power Save Mode

Power Save Mode must be enabled using the following steps.

1. Disable the LCD bias power using GPO0.

Note
The S5U13708B00B uses GPO0 to control the LCD bias power supplies. Your system design may vary.

2. Wait for the LCD bias power supply to discharge. The discharge time must be based on the time specified in the LCD panel specification.
3. Enable Power Save Mode - set REG[A0h] bit 0 to 1.
4. At this time, the LCD pixel clock source may be disabled (Optional).
5. Optionally, when the Memory Controller Power Save Status bit (REG[A0h] bit 3) returns a 1, the Memory Clock source may be safely shut down.

6.5 Disabling Power Save Mode

Power Save Mode must be disabled using the following steps.

1. If the Memory Clock source is shut down, it must be started and the Memory Controller Power Save Status bit must return a 0. **If the pixel clock source is disabled, it must be started before step 2.**
2. Disable Power Save Mode - set REG[A0h] bit 0 to 0.
3. Wait for the LCD bias power supply to charge. The charge time must be based on the time specified in the LCD panel specification.
4. Enable the LCD bias power using GPO0.

Note
The S5U13708B00B uses GPO0 to control the LCD bias power supplies. Your system design may vary.

7 SwivelView™

Most computer displays operate in landscape mode. In landscape mode the display is typically wider than it is high. For example, a display size of 320x240 is 320 pixels wide and 240 lines high.

SwivelView rotates the display image counter-clockwise in ninety degree increments. Rotating the image on a 320x240 display by 90 or 270 degrees yields a display that is now 240 pixels wide and 320 lines high.

The S1D13708 provides hardware support for SwivelView in all color depths (1, 2, 4, 8 and 16 bpp).

For further details on the SwivelView feature, see the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

7.1 SwivelView Registers

The following registers control the SwivelView feature.

Special Effects Register
REG[71h] Read/Write

Display Data Word Swap	Display Data Byte Swap	n/a	PIP+ Window Enable	n/a		SwivelView Mode Select Bits 1-0	
7	6	5	4	3	2	1	0

SwivelView Mode Select

The SwivelView modes are selected using the SwivelView Mode Select Bits[1:0] (bits 1-0). The combinations of these bits provide the following rotations:

Table 7-1: SwivelView™ Mode Select Options

SwivelView Mode Select Bits	SwivelView Orientation
00	0° (Normal)
01	90°
10	180°
11	270°

Main Window Display Start Address Register 0
REG[74h] Read/Write

7	6	5	4	3	2	1	0
Main window Display Start Address Bits 7-0							

Main Window Display Start Address Register 1
REG[75h] Read/Write

7	6	5	4	3	2	1	0
Main window Display Start Address Bits 15-8							

Main Window Display Start Address Register 2
REG[76h] Read/Write

7	6	5	4	3	2	1	0
n/a							Main window Display Start Address Bit 16

Main Window Display Start Address

The Main Window Display Start Address register represents a DWORD address which points to the start of the main window image in the display buffer. An address of 0 is the start of the display buffer. For the following SwivelView mode descriptions, the *desired byte address* is the starting display address for the main window image.

In SwivelView 0°, program the start address
= desired byte address ÷ 4

In SwivelView 90°, program the start address
= ((desired byte address + (panel height × bpp ÷ 8)
+ ((4 - (panel height × bpp ÷ 8)) & 03h)) ÷ 4) - 1

In SwivelView 180°, program the start address
= ((desired byte address + (Main Window Stride × (panel height − 1))
+ (panel width × bpp ÷ 8) + ((4 - (panel width × bpp ÷ 8)) & 03h)) ÷ 4) - 1

In SwivelView 270°, program the start address
= (desired byte address + ((panel width - 1) × Main Window Stride)) ÷ 4

Note
Truncate all fractional values before writing to the address registers.

Note
SwivelView 0° and 180° require the panel width to be a multiple of 32 ÷ bits-per-pixel. SwivelView 90° and 270° require the panel height to be a multiple of 32 ÷ bits-per-pixel. If this is not possible, refer to Section 7.3, “Limitations” .

Main Window Line Address Offset Register 0
REG[78h] Read/Write

7	6	5	4	3	2	1	0
Main window Line Address Offset Bits 7-0							

Main Window Line Address Offset Register 1
REG[79h] Read/Write

7	6	5	4	3	2	1	0
n/a						Main window Line Address Offset Bits 9-8	

Main Window Line Address Offset

The Main Window Line Address Offset register indicates the number of dwords per line in the main window image.

For SwivelView 0° and 180°, the image width must be at least the panel width. For SwivelView 90° and 270°, the image width must be at least the panel height. In addition, the image width must be a multiple of 32 ÷ bpp. If the image width is not such a multiple, a slightly larger width must be chosen (see Section 7.3, "Limitations").

Panel width and *panel height* refer to the physical panel dimensions in pixels. *Stride* is the number of bytes required for one line of the image; the offset register represents the stride in DWORD steps.

Main Window Stride = image width × bpp ÷ 8

Note
Image width can be larger than panel width (or panel height, for SwivelView 90° or 270°).

number of dwords per line = image width ÷ (32 ÷ bpp)

7.2 Examples

Example 1: In SwivelView 0° (normal) mode, program the main window registers for a 320x240 panel at a color depth of 4 bpp.

1. Determine the main window display start address.
   The main window is typically placed at the start of display memory which is at display address 0.

   main window display start address register
   = desired byte address ÷ 4
   = 0

   Program the Main Window Display Start Address registers.
   REG[76h],REG[75h],REG[74h] are set to 000000h.

2. Determine the main window line address offset.

   number of dwords per line
   = image width ÷ (32 ÷ bpp)
   = 320 ÷ (32 ÷ 4)
   = 40
   = 28h

   Program the Main Window Line Address Offset registers. REG[79h],REG[78h] are set to 0028h.

Example 2: In SwivelView 90° mode, program the main window registers for a 320x240 panel at a color depth of 4 bpp.

1. Determine the main window display start address.
   The main window is typically placed at the start of display memory, which is at display address 0.

   main window display start address register
   = ((desired byte address + (panel height × bpp ÷ 8)
   + ((4 - (panel height × bpp ÷ 8)) & 03h)) ÷ 4) - 1
   = ((0 + (240 × 4 ÷ 8) + ((4 - (240 × 4 ÷ 8)) & 03h)) ÷ 4) - 1
   = 29
   = 1Dh

   Program the Main Window Display Start Address registers.
   REG[76h],REG[75h],REG[74h] are set to 00001Dh.

2. Determine the main window line address offset.

   number of dwords per line
   = image width ÷ (32 ÷ bpp)
   = 240 ÷ (32 ÷ 4)
   = 30
   = 1Eh

   Program the Main Window Line Address Offset registers. REG[79h],REG[78h] are set to 001Eh.

Example 3: In SwivelView 180° mode, program the main window registers for a 320x240 panel at a color depth of 4 bpp.

1. Determine the main window display start address.
   The main window is typically placed at the start of display memory which is at display address 0.

   Main Window Stride
   = image width × bpp ÷ 8
   = 320 × 4 ÷ 8
   = 160
   = A0h

   main window display start address register
   = ((desired byte address + (Main Window Stride × (panel height − 1))
   + (panel width × bpp ÷ 8) + ((4 - (panel width × bpp ÷ 8)) & 03h)) ÷ 4) - 1
   = ((0+(160 × (240 − 1)) + (320 × 4 ÷ 8) + ((4 - (320 × 4 ÷ 8))& 03h)) ÷ 4) - 1
   = 9599
   = 257Fh

   Program the Main Window Display Start Address registers.
   REG[76h],REG[75h],REG[74h] are set to 00257Fh.

2. Determine the main window line address offset.

   number of dwords per line
   = image width ÷ (32 ÷ bpp)
   = 320 ÷ (32 ÷ 4)
   = 40
   = 28h

   Program the Main Window Line Address Offset registers. REG[79h],REG[78h] are set to 0028h.

Example 4: In SwivelView 270° mode, program the main window registers for a 320x240 panel at a color depth of 4 bpp.

1. Determine the main window display start address.
   The main window is typically placed at the start of display memory, which is at display address 0.

   Main Window Stride
   = image width × bpp ÷ 8
   = 240 × 4 ÷ 8
   = 120
   = 78h

   main window display start address register
   = (desired byte address + ((panel width - 1) × Main Window Stride)) ÷ 4
   = (0 + ((320 - 1) × 120)) ÷ 4
   = 9570
   = 2562h

   Program the Main Window Display Start Address registers.
   REG[76h],REG[75h],REG[74h] are set to 002562h.

2. Determine the main window line address offset.

   number of dwords per line
   = image width ÷ (32 ÷ bpp)
   = 240 ÷ (32 ÷ 4)
   = 30
   = 1Eh

   Program the Main Window Line Address Offset registers. REG[79h],REG[78h] are set to 001Eh.

7.3 Limitations

7.3.1 SwivelView 0° and 180°

In SwivelView 0° and 180°, the Main Window Line Address Offset registers (REG[79h],REG[78h]) require the ***panel width*** to be a multiple of 32 ÷ bits-per-pixel. If this is not the case, then the Main Window Line Address Offset registers must be programmed to a longer line which meets this requirement. This longer line creates a virtual image where the width is *main window line address offset registers* × *32* ÷ *bits-per-pixel*. In SwivelView 0°, this virtual image should be drawn in display memory as left justified, and in SwivelView 180°, this virtual image should be drawn in display memory as right justified. A left-justified image is one drawn in display memory such that each of the image's lines only use the left most portion of the line width defined by the line address offset register (i.e. starting at horizontal position 0). A right-justified image is one drawn in display memory such that each of the image's lines only use the right most portion of the line width defined by the line address offset registers (i.e. starting at a non-zero horizontal position which is the virtual width - image width).

7.3.2 SwivelView 90° and 270°

In SwivelView 90° and 270°, the Main Window Line Address Offset registers (REG[79h],REG[78h]) require the ***panel height*** to be a multiple of 32 ÷ bits-per-pixel. If this is not the case, then the Main Window Line Address Offset registers must be programmed to a longer line which meets this requirement. This longer line creates a virtual image whose width is *main window line address offset registers* × *32* ÷ *bits-per-pixel*. In SwivelView 270°, this virtual image should be drawn in display memory as left justified, and in SwivelView 90°, this virtual image should be drawn in display memory as right justified. A left-justified image is one drawn in display memory such that each of the image's lines only use the left most portion of the line width defined by the line address offset register (i.e. starting at horizontal position 0). A right-justified image is one drawn in display memory such that each of the image's lines only use the right most portion of the line width defined by the line address offset register (i.e. starting at a non-zero horizontal position which is the virtual width - image width).

8 Picture-In-Picture Plus

8.1 Concept

Picture-in-Picture Plus (PIP+) enables a secondary window (or PIP+ window) within the main display window. The PIP+ window may be positioned anywhere within the virtual display and is controlled through the PIP+ Window control registers (REG[7Ch] through REG[91h]). The PIP+ window retains the same color depth and SwivelView orientation as the main window.

The following diagram shows an example of a PIP+ window within a main window.

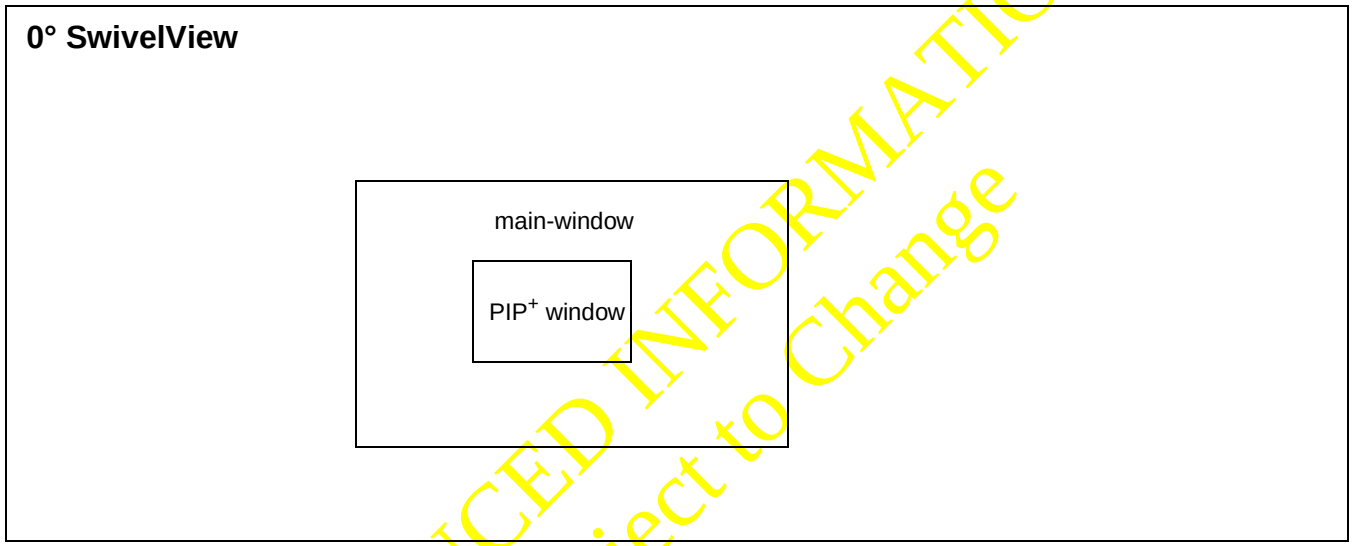


Figure 8-1: Picture-in-Picture Plus with SwivelView disabled

8.2 Registers

The following registers control the Picture-In-Picture Plus feature.

Special Effects Register
REG[71h] Read/Write

Display Data Word Swap	Display Data Byte Swap	n/a	PIP+ Window Enable	n/a		SwivelView Mode Select Bits 1-0	
7	6	5	4	3	2	1	0

PIP+ Window Enable

The PIP+ Window Enable bit enables a PIP+ window within the main window. The location of the PIP+ window within the landscape window is determined by the PIP+ X Position registers (REG[84h],REG[85h],REG[8Ch],REG[8Dh]) and PIP+ Y Position registers (REG[88h],REG[89h],REG[90h],REG[91h]). The PIP+ window has its own Display Start Address registers (REG[7Eh],REG[7Dh],REG[7Ch]) and Line Address Offset registers (REG[81h],REG[80h]). The PIP+ window shares the same color depth and SwivelView™ orientation as the main window.

PIP+ Window Display Start Address Register 0
REG[7C] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window Display Start Address Bits 7-0							

PIP+ Window Display Start Address Register 1
REG[7Dh] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window Display Start Address Bits 15-8							

PIP+ Window Display Start Address Register 2
REG[7Eh] Read/Write

7	6	5	4	3	2	1	0
n/a							PIP+ Window Display Start Address Bit 16

PIP+ Display Start Address

The PIP+ Display Start Address register is a DWORD which represents an address that points to the start of the PIP+ window image in the display buffer. An address of 0 is the start of the display buffer. For the following PIP+ descriptions, the *desired byte address* is the starting display address for the PIP+ window image.

In SwivelView 0°, program the start address
= desired byte address ÷ 4

In SwivelView 90°, program the start address
= ((desired byte address + (PIP+ width × bpp ÷ 8)
+ ((4 - (PIP+ width × bpp ÷ 8)) & 03h)) ÷ 4) - 1

In SwivelView 180°, program the start address
= ((desired byte address + (PIP+ Stride × (PIP+ height − 1))
+ (PIP+ width × bpp ÷ 8) + ((4 - (PIP+ width × bpp ÷ 8)) & 03h)) ÷ 4) - 1

In SwivelView 270°, program the start address
= (desired byte address + ((PIP+ height - 1) × PIP+ Stride)) ÷ 4

Note
Truncate all fractional values before writing to the address registers.

Note
SwivelView 0° and 180° require the PIP+ width to be a multiple of 32 ÷ bits-per-pixel. SwivelView 90° and 270° require the PIP+ height to be a multiple of 32 ÷ bits-per-pixel. If this is not possible, refer to Section 8.4, “Limitations” .

PIP+ Window Line Address Offset Register 0
REG[80h] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window Line Address Offset Bits 7-0							

PIP+ Window Line Address Offset Register 1
REG[81h] Read/Write

7	6	5	4	3	2	1	0
n/a						PIP+ Window Line Address Offset Bits 9-8	

PIP+ Line Address Offset

The PIP+ Line Address Offset register indicates the number of dwords per line in the PIP+ window image.

The image width must be a multiple of 32 ÷ bpp. If the image width is not such a multiple, a slightly larger width must be chosen (see Section 8.4, “Limitations”).

PIP+ width and *PIP+ height* refer to the PIP+ dimensions as seen in SwivelView 0° (landscape mode). *Stride* is the number of bytes required for one line of the image; the offset register represents the stride in DWORD steps.

PIP+ Stride = image width × bpp ÷ 8

For SwivelView 0° and 180°,

PIP+ Width=((REG[8Dh],REG[8Ch]) - (REG[85h],REG[84h]) + 1) × (32 ÷ bpp)

PIP+ Height=(REG[91h],REG[90h]) - (REG[89h],REG[88h]) + 1

For SwivelView 90° and 270°,

PIP+ Width=((REG[91h],REG[90h]) - (REG[89h],REG[88h]) + 1) × (32 ÷ bpp)

PIP+ Height=(REG[8Dh],REG[8Ch]) - (REG[85h],REG[84h]) + 1

Note
Image width can be larger than PIP+ width (or PIP+ height, for SwivelView 90° or 270°).

number of dwords per line = image width ÷ (32 ÷ bpp)

PIP+ Window X Start Position Register 0
REG[84h] Read/Write

PIP+ Window X Start Position Bits 7-0							
7	6	5	4	3	2	1	0

PIP+ Window X Start Position Register 1
REG[85h] Read/Write

n/a						PIP+ Window X Start Position Bits 9-8	
7	6	5	4	3	2	1	0

PIP+ X Start Position

The PIP+ X Start Position bits determine the horizontal position of the start of the PIP+ window in 0° and 180° SwivelView orientations. These bits determine the vertical start position in 90° and 270° SwivelView. For further information on defining the value of the X Start Position, see Section 8.3, “Picture-In-Picture-Plus Examples” on page 48.

The register also increments differently based on the SwivelView orientation. For 0° and 180° SwivelView the X Start Position is incremented by *X* pixels where *X* is relative to the current color depth. For 90° and 270° SwivelView the X Start Position is incremented in 1 line increments.

Table 8-1: 32-bit Address Increments for Color Depth

Bits-per-pixel (Color Depth)	Pixel Increment (X)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

In SwivelView 0°, these bits set the horizontal coordinates (x) of the PIP+ windows’s left edge. Increasing x moves the left edge towards the right in steps of (32 ÷ bits-per-pixel) (see Table 8-1:, “32-bit Address Increments for Color Depth”). The horizontal coordinates start at pixel 0.

Program the PIP+ Window X Start Position so that
PIP+ Window X Start Position = x ÷ (32 ÷ bits-per-pixel)

Note
Truncate the fractional part of the above equation.

In SwivelView 90°, these bits set the vertical coordinates (y) of the PIP+ window's top edge. Increasing y moves the top edge downward in 1 line steps. The vertical coordinates start at line 0.

Program the PIP+ Window X Start Position so that
PIP+ Window X Start Position = y

In SwivelView 180°, these bits set the horizontal coordinates (x) of the PIP+ window's right edge. Increasing x moves the right edge towards the right in steps of (32 ÷ bits-per-pixel) (see Table 8-2:, "32-bit Address Increments for Color Depth"). The horizontal coordinates start at pixel 0.

Program the PIP+ Window X Start Position so that
PIP+ Window X Start Position = (panel width - x - 1) ÷ (32 ÷ bits-per-pixel)

Note
Truncate the fractional part of the above equation.

In SwivelView 270°, these bits set the vertical coordinates (y) of the PIP+ window's bottom edge. Increasing y moves the bottom edge downwards in 1 line steps. The vertical coordinates start at line 0.

Program the PIP+ Window X Start Position so that
PIP+ Window X Start Position = panel width - y - 1

PIP+ Window Y Start Position Register 0
REG[88h] Read/Write

PIP+ Window Y Start Position Bits 7-0							
7	6	5	4	3	2	1	0

PIP+ Window Y Start Position Register 1
REG[89h] Read/Write

n/a						PIP+ Window Y Start Position Bits 9-8	
7	6	5	4	3	2	1	0

PIP+ Y Start Position

The PIP+ Y Start Position bits determine the vertical start position of the PIP+ window in 0° and 180° SwivelView orientations. These bits determine the horizontal start position in 90° and 270° SwivelView. For further information on defining the value of the Y Start Position, see Section 8.3, “Picture-In-Picture-Plus Examples” on page 48.

The register also increments differently based on the SwivelView orientation. For 0° and 180° SwivelView the Y Start Position is incremented in 1 line increments. For 90° and 270° SwivelView the Y Start Position is incremented by *Y* pixels where *Y* is relative to the current color depth.

Table 8-2: 32-bit Address Increments for Color Depth

Bits-Per-Pixel (Color Depth)	Pixel Increment (Y)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

In SwivelView 0°, these bits set the vertical coordinates (y) of the PIP+ windows’s top edge. Increasing y moves the top edge downwards in 1 line steps. The vertical coordinates start at line 0.

Program the PIP+ Window Y Start Position so that
PIP+ Window Y Start Position = y

In SwivelView 90°, these bits set the horizontal coordinates (x) of the PIP+ window’s right edge. Increasing x moves the right edge towards the right in steps of (32 ÷ bits-per-pixel) (see Table 8-2:, “32-bit Address Increments for Color Depth”). The horizontal coordinates start at pixel 0.

Program the PIP+ Window Y Start Position so that
PIP+ Window Y Start Position = (panel height - x - 1) ÷ (32 ÷ bits-per-pixel)

Note
Truncate the fractional part of the above equation.

In SwivelView 180°, these bits set the vertical coordinates (y) of the PIP+ window's bottom edge. Increasing y moves the bottom edge downwards in 1 line steps. The vertical coordinates start at line 0.

Program the PIP+ Window Y Start Position so that
PIP+ Window Y Start Position = panel height - y - 1

In SwivelView 270°, these bits set the horizontal coordinates (x) of the PIP+ window's left edge. Increasing x moves the left edge towards the right in steps of (32 ÷ bits-per-pixel) (see Table 8-2:, "32-bit Address Increments for Color Depth"). The horizontal coordinates start at pixel 0.

Program the PIP+ Window Y Start Position so that
PIP+ Window Y Start Position = x ÷ (32 ÷ bits-per-pixel)

Note
Truncate the fractional part of the above equation.

PIP+ Window X End Position Register 0
REG[8Ch] Read/Write

PIP+ Window X End Position Bits 7-0							
7	6	5	4	3	2	1	0

PIP+ Window X End Position Register 1
REG[8Dh] Read/Write

n/a						PIP+ Window X End Position Bits 9-8	
7	6	5	4	3	2	1	0

PIP+ X End Position

The PIP+ X End Position bits determine the horizontal end of the PIP+ window in 0° and 180° SwivelView orientations. These bits determine the vertical end position in 90° and 270° SwivelView. For further information on defining the value of the X End Position, see Section 8.3, “Picture-In-Picture-Plus Examples” on page 48.

This register also increments differently based on the SwivelView orientation. For 0° and 180° SwivelView the X End Position is incremented by *X* pixels where *X* is relative to the current color depth. For 90° and 270° SwivelView the X End Position is incremented in 1 line increments.

Table 8-3: 32-bit Address Increments for Color Depth

Bits-Per-Pixel (Color Depth)	Pixel Increment (X)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

In SwivelView 0°, these bits set the horizontal coordinates (x) of the PIP+ window’s right edge. Increasing x moves the right edge towards the right in steps of 32 ÷ bits-per-pixel (see Table 8-3:, “32-bit Address Increments for Color Depth”). The horizontal coordinates start at pixel 0.

Program the PIP+ Window X End Position so that
PIP+ Window X End Position = x ÷ (32 ÷ bits-per-pixel)

Note
Truncate the fractional part of the above equation.

In SwivelView 90°, these bits set the vertical coordinates (y) of the PIP+ window’s bottom edge. Increasing y moves the bottom edge downward in 1 line steps. The vertical coordinates start at line 0.

Program the PIP+ Window X End Position so that
PIP+ Window X End Position = y

In SwivelView 180°, these bits set the horizontal coordinates (x) of the PIP+ window's left edge. Increasing x moves the left edge towards the right in steps of 32 ÷ bits-per-pixel (see Table 8-3:, "32-bit Address Increments for Color Depth"). The horizontal coordinates start at pixel 0.

Program the PIP+ Window X End Position so that
PIP+ Window X End Position = (panel width - x - 1) ÷ (32 ÷ bits-per-pixel)

Note
Truncate the fractional part of the above equation.

In SwivelView 270°, these bits set the vertical coordinates (y) of the PIP+ window's top edge. Increasing y moves the top edge downwards in 1 line steps. The vertical coordinates start at line 0.

Program the PIP+ Window X End Position so that
PIP+ Window X End Position = panel width - y - 1

PIP+ Window Y End Position Register 0
REG[90h] Read/Write

7	6	5	4	3	2	1	0
PIP+ Window Y End Position Bits 7-0							

PIP+ Window Y End Position Register 1
REG[91h] Read/Write

7	6	5	4	3	2	1	0
n/a						PIP+ Window Y End Position Bits 9-8	

PIP+ Y End Position

The PIP+ Y End Position bits determine the vertical end position of the PIP+ window in 0° and 180° SwivelView orientations. These bits determine the horizontal end position in 90° and 270° SwivelView. For further information on defining the value of the Y End Position, see Section 8.3, "Picture-In-Picture-Plus Examples" on page 48.

The register also increments differently based on the SwivelView orientation. For 0° and 180° SwivelView the Y End Position is incremented in 1 line increments. For 90° and 270° SwivelView the Y End Position is incremented by *Y* pixels where *Y* is relative to the current color depth.

Table 8-4: 32-bit Address Increments for Color Depth

Bits-Per-Pixel (Color Depth)	Pixel Increment (Y)
1 bpp	32
2 bpp	16
4 bpp	8
8 bpp	4
16 bpp	2

In SwivelView 0°, these bits set the vertical coordinates (y) of the PIP+ windows's bottom edge. Increasing y moves the bottom edge downwards in 1 line steps. The vertical coordinates start at line 0.

Program the PIP+ Window Y End Position so that
PIP+ Window Y End Position = y

In SwivelView 90°, these bits set the horizontal coordinates (x) of the PIP+ window's left edge. Increasing x moves the left edge towards the right in steps of (32 ÷ bits-per-pixel) (see Table 8-4:, "32-bit Address Increments for Color Depth"). The horizontal coordinates start at pixel 0.

Program the PIP+ Window Y End Position so that
PIP+ Window Y End Position = (panel height - x - 1) ÷ (32 ÷ bits-per-pixel)

Note
Truncate the fractional part of the above equation.

In SwivelView 180°, these bits set the vertical coordinates (y) of the PIP+ window's top edge. Increasing y moves the top edge downwards in 1 line steps. The vertical coordinates start at line 0.

Program the PIP+ Window Y End Position so that
PIP+ Window Y End Position = panel height - y - 1

In SwivelView 270°, these bits set the horizontal coordinates (x) of the PIP+ window's right edge. Increasing x moves the right edge towards the right in steps of (32 ÷ bits-per-pixel) (see Table 8-4:, "32-bit Address Increments for Color Depth"). The horizontal coordinates start at pixel 0.

Program the PIP+ Window Y End Position so that
PIP+ Window Y End Position = x ÷ (32 ÷ bits-per-pixel)

Note
Truncate the fractional part of the above equation.

8.3 Picture-In-Picture-Plus Examples

8.3.1 SwivelView 0° (Landscape Mode)

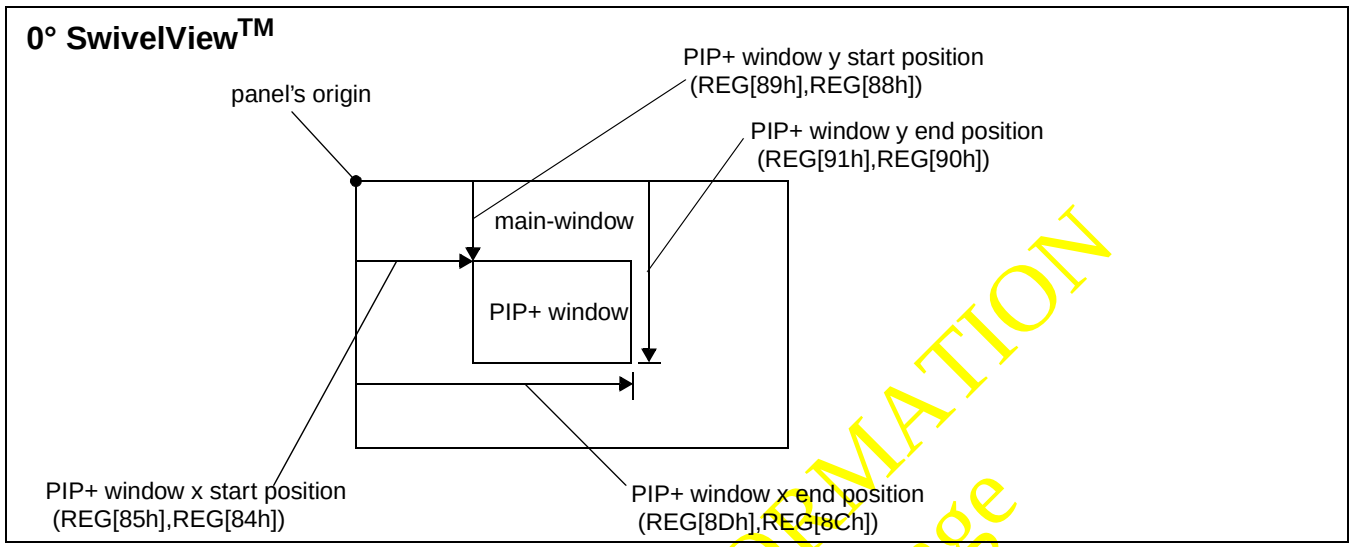


Figure 8-2: Picture-in-Picture Plus with SwivelView disabled

SwivelView 0° (or landscape) is a mode in which both the main and PIP+ window are non-rotated. The images for each window are typically placed consecutively, with the main window image starting at address 0 and followed by the PIP+ window image. In addition, both images must start at addresses which are dword-aligned (the last two bits of the starting address must be 0).

Note

It is possible to use the same image for both the main window and PIP+ window. To do so, set the PIP+ Line Address Offset register (REG[81h],REG[80h]) to the same value as the Main Window Line Address Offset register (REG[79h],REG[78h]).

Example 5: Program the PIP+ window registers for a 320x240 panel at 4 bpp, with the PIP+ window positioned at (80, 60) with a width of 160 and a height of 120.

1. Determine the value for the PIP+ Window X Positions and PIP+ Window Y Positions registers. Let the top left corner of the PIP+ window be (x1, y1), and let the bottom right corner be (x2, y2), where x2 = x1 + width - 1 and y2 = y1 + height - 1. The PIP+ Window X Positions register sets the horizontal coordinates of the PIP+ window's top left and bottom right corners. The PIP+ Window Y Positions register sets the vertical coordinates of the PIP+ window's top left and bottom right corners.

   The required values are calculated as follows:

   X Start Position
   = x1 ÷ (32 ÷ bpp)
   = 80 ÷ (32 ÷ 4)
   = 10
   = 0Ah

   Y Start Position
   = y1
   = 60
   = 3Ch

   X End Position
   = x2 ÷ (32 ÷ bpp)
   = (80 + 160 - 1) ÷ (32 ÷ 4)
   = 29.875
   = 1Dh (truncated fractional part)

   Y End Position
   = y2
   = 60 + 120 - 1
   = 179
   = B3h

2. Program the PIP+ Window X Positions register with the X Start Position in (REG[85h],REG[84h]) and the X End Position in (REG[8Dh],REG[8Ch]).
   REG[84h]=0Ah
   REG[85h]=00h
   REG[8Ch]=1Dh
   REG[8Dh]=00h
   Program the PIP+ Window Y Positions register with the Y Start Position in (REG[89h],REG[88h]) and the Y End Position in (REG[91h],REG[90h]).
   REG[88h]=3Ch
   REG[89h]=00h
   REG[90h]=B3h
   REG[91h]=00h

   Note that the values of REG[84h] through REG[91h] do not go into effect until

after REG[91h] is written.

Due to truncation, the dimensions of the PIP+ window may have changed. Recalculate the PIP+ window width and height below:
PIP+ Width
= ((REG[8Dh],REG[8Ch]) - (REG[85h],REG[84h]) + 1) × (32 ÷ bpp)
= (1Dh - 0Ah + 1) × (32 ÷ 4)
= 160 pixels

PIP Height
= (REG[91h],REG[90h]) - (REG[89h],REG[88h]) + 1
= B3h - 3Ch + 1
= 120 lines

3. Determine the PIP+ display start address.
The main window image must take up 320 x 240 pixels × bpp ÷ 8 = 9600h bytes. If the main window starts at address 0h, the PIP+ window can start at 9600h.

PIP+ display start address
= desired byte address ÷ 4
= 9600h ÷ 4
= 2580h.

Program the PIP+ Display Start Address registers.
REG[7Ch]=80h
REG[7Dh]=25h
REG[7Eh]=00h

4. Determine the PIP+ line address offset.

number of dwords per line
= image width ÷ (32 ÷ bpp)
= 160 ÷ (32 ÷ 4)
= 20
= 14h

Program the PIP+ Line Address Offset registers.
REG[80h]=14h
REG[81h]=00h.

5. Enable the PIP+ window.

Program the PIP+ Window Enable bit.
REG[71h] bit 4 is set to 1.

8.3.2 SwivelView 90°

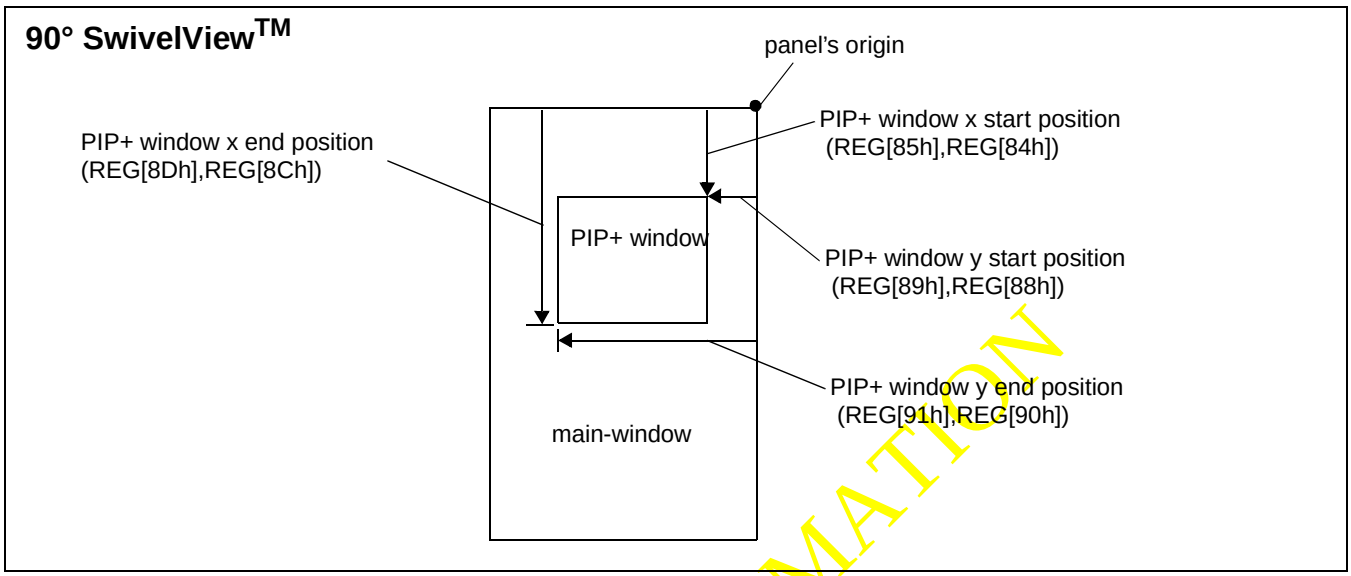


Figure 8-3: Picture-in-Picture Plus with SwivelView 90° enabled

SwivelView 90° is a mode in which both the main and PIP+ windows are rotated 90° counter-clockwise when shown on the panel. The images for each window are typically placed consecutively, with the main window image starting at address 0 and followed by the PIP+ window image. In addition, both images must start at addresses which are dword-aligned (the last two bits of the starting address must be 0).

Note

It is possible to use the same image for both the main window and PIP+ window. To do so, set the PIP+ Line Address Offset register (REG[81h],REG[80h]) to the same value as the Main Window Line Address Offset register (REG[79h],REG[78h]).

Example 6: In SwivelView 90°, program the PIP+ window registers for a 320x240 panel at 4 bpp, with the PIP+ window positioned at SwivelView 90° coordinates (60, 80) with a width of 120 and a height of 160.

1. Determine the value for the PIP+ Window X Positions and PIP+ Window Y Positions registers. Let the top left corner of the PIP+ window be (x1, y1), and let the bottom right corner be (x2, y2), where x2 = x1 + width - 1 and y2 = y1 + height - 1. The PIP+ Window X Positions register sets the vertical coordinates of the PIP+ window's top right and bottom left corners. The PIP+ Window Y Positions register sets the horizontal coordinates of the PIP+ window's top right and bottom left corners.

   The required values are calculated as follows:

   X Start Position
   = y1
   = 80
   = 50h

   Y Start Position
   = (panel height - x2 - 1) ÷ (32 ÷ bpp)
   = (240 - (60 + 120 - 1) - 1) ÷ (32 ÷ 4)
   = 7.5
   = 07h (truncated fractional part)

   X End Position
   = y2
   = 80 + 160 - 1
   = 239
   = EFh

   Y End Position
   = (panel height - x1 - 1) ÷ (32 ÷ bpp)
   = (240 - 60 - 1) ÷ (32 ÷ 4)
   = 22.375
   = 16h (truncated fractional part)

2. Program the PIP+ Window X Positions register with the X Start Position in (REG[85h],REG[84h]) and the X End Position in (REG[8Dh],REG[8Ch]).
   REG[84h]=50h
   REG[85h]=00h
   REG[8Ch]=EFh
   REG[8Dh]=00h
   Program the PIP+ Window Y Positions register with the Y Start Position in (REG[89h],REG[88h]) and the Y End Position in (REG[91h],REG[90h]).
   REG[88h]=07h
   REG[89h]=00h
   REG[90h]=16h
   REG[91h]=00h

Note that the values of REG[84h] through REG[91h] do not go into effect until after REG[91h] is written.

Due to truncation, the dimensions of the PIP+ window may have changed. Recalculate the PIP+ window width and height below:
PIP Width
= ((REG[91h],REG[90h]) - (REG[89h],REG[88h]) + 1) × (32 ÷ bpp)
= (16h - 07h + 1) × (32 ÷ 4)
= 128 pixels **(note that this is different from the desired width)**

PIP+ Height
= (REG[8Dh],REG[8Ch]) - (REG[85h],REG[84h]) + 1
= EFh - 50h + 1
= 160 lines

3. Determine the PIP+ display start address.
The main window image must take up 320 x 240 pixels × bpp ÷ 8 = 9600h bytes. If the main window starts at address 0h, then the PIP+ window can start at 9600h.

PIP+ display start address
= ((desired byte address + (PIP+ width × bpp ÷ 8)
+ ((4 - (PIP+ width × bpp ÷ 8)) & 03h)) ÷ 4) - 1
= ((9600h + (128 × 4 ÷ 8) + ((4 - (128 × 4 ÷ 8)) & 03h)) ÷ 4) - 1
= 9615
= 258Fh

Program the PIP+ Display Start Address registers.
REG[7Ch]=8Fh
REG[7Dh]=25h
REG[7Eh]=00h

4. Determine the PIP+ line address offset.

number of dwords per line
= image width ÷ (32 ÷ bpp)
= 128 ÷ (32 ÷ 4)
= 16
= 10h

Program the PIP+ Line Address Offset registers.
REG[80h]=10h
REG[81h]=00h.

5. Enable the PIP+ window.

Program the PIP+ Window Enable bit.
REG[71h] bit 4 is set to 1.

8.3.3 SwivelView 180°

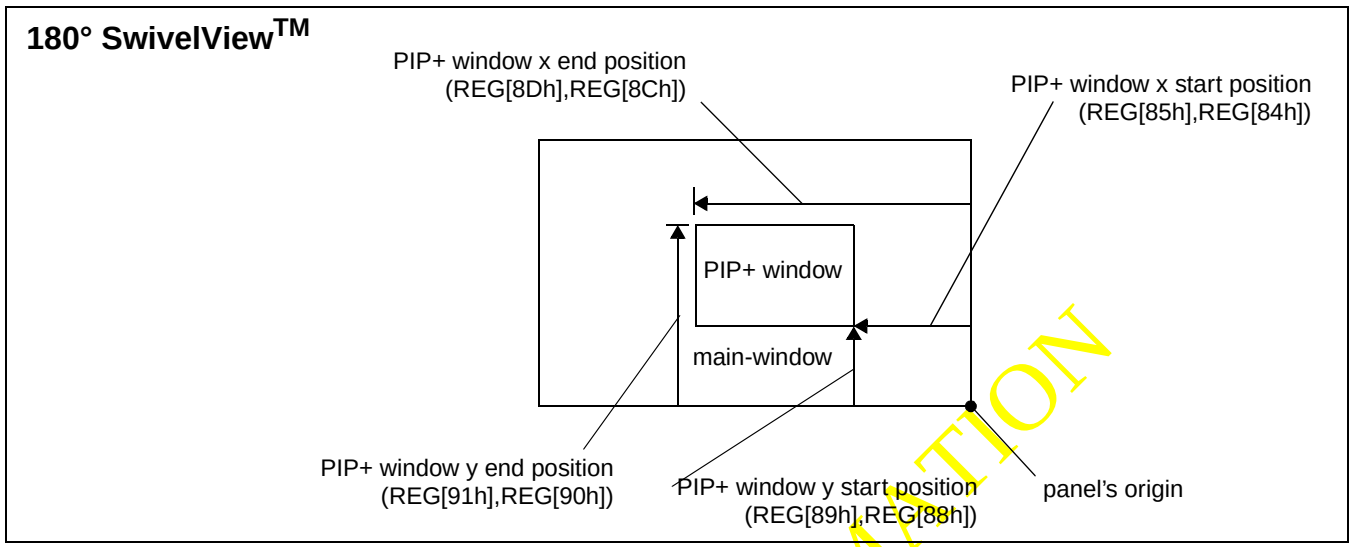


Figure 8-4: Picture-in-Picture Plus with SwivelView 180° enabled

SwivelView 180° is a mode in which both the main and PIP+ windows are rotated 180° counter-clockwise when shown on the panel. The images for each window are typically placed consecutively, with the main window image starting at address 0 and followed by the PIP+ window image. In addition, both images must start at addresses which are dword-aligned (the last two bits of the starting address must be 0).

Note

It is possible to use the same image for both the main window and PIP+ window. To do so, set the PIP+ Line Address Offset register (REG[81h],REG[80h]) to the same value as the Main Window Line Address Offset register (REG[79h],REG[78h]).

Example 7: In SwivelView 180°, program the PIP+ window registers for a 320x240 panel at 4 bpp, with the PIP+ window positioned at SwivelView 180° coordinates (80, 60) with a width of 160 and a height of 120.

1. Determine the value for the PIP+ Window X Positions and PIP+ Window Y Positions registers. Let the top left corner of the PIP+ window be (x1, y1), and let the bottom right corner be (x2, y2), where x2 = x1 + width - 1 and y2 = y1 + height - 1. The PIP+ Window X Positions register sets the horizontal coordinates of the PIP+ window's bottom right and top left corner. The PIP+ Window Y Positions register sets the vertical coordinates of the PIP+ window's bottom right and top left corner.

   The required values are calculated as follows:

   X Start Position
   = (panel width - x2 - 1) ÷ (32 ÷ bpp)
   = (320 - (80 + 160 - 1) - 1) ÷ (32 ÷ 4)
   = 10
   = 0Ah
   Y Start Position
   = panel height - y2 - 1
   = 240 - (60 + 120 - 1) - 1
   = 60
   = 3Ch
   X End Position
   = (panel width - x1 - 1) ÷ (32 ÷ bpp)
   = (320 - 80 - 1) ÷ (32 ÷ 4)
   = 29.875
   = 1Dh (truncated fractional part)
   Y End Position
   = panel height - y1 - 1
   = 240 - 60 - 1
   = 179
   = B3h

2. Program the PIP+ Window X Positions register with the X Start Position in (REG[85h],REG[84h]) and the X End Position in (REG[8Dh],REG[8Ch]).
   REG[84h]=0Ah
   REG[85h]=00h
   REG[8Ch]=1Dh
   REG[8Dh]=00h
   Program the PIP+ Window Y Positions register with the Y Start Position in (REG[89h],REG[88h]) and the Y End Position in (REG[91h],REG[90h]).
   REG[88h]=3Ch
   REG[89h]=00h
   REG[90h]=B3h
   REG[91h]=00h

   Note that the values of REG[84h] through REG[91h] do not go into effect until after REG[91h] is written.

Due to truncation, the dimensions of the PIP+ window may have changed. Recalculate the PIP+ window width and height below:

PIP+ Width

= ((REG[8Dh],REG[8Ch]) - (REG[85h],REG[84h]) + 1) × (32 ÷ bpp)
= (1Dh - 0Ah + 1) × (32 ÷ 4)
= 160 pixels

PIP Height

= (REG[91h],REG[90h]) - (REG[89h],REG[88h]) + 1
= B3h - 3Ch + 1
= 120 lines

3. Determine the PIP+ display start address.
The main window image must take up 320 x 240 pixels × bpp ÷ 8 = 9600h bytes. If the main window starts at address 0h, then the PIP+ window can start at 9600h.

PIP+ Stride

= image width × bpp ÷ 8
= 160 × 4 ÷ 8
= 80
= 50h

PIP+ display start address

= ((desired byte address + (PIP+ Stride × (PIP+ height - 1))
+ (PIP+ width × bpp ÷ 8) + ((4 - (PIP width × bpp ÷ 8)) & 03h)) ÷ 4) - 1
= ((9600h + (80 × (120 - 1)) + (160 × 4 ÷ 8) + ((4 - (160 × 4 ÷ 8))&03h)) ÷ 4) - 1
= 11999
= 2EDFh

Program the PIP+ Display Start Address registers.

REG[7Ch]=DFh
REG[7Dh]=2Eh
REG[7Eh]=00h

4. Determine the PIP+ line address offset.

   number of dwords per line
   = image width ÷ (32 ÷ bpp)
   = 160 ÷ (32 ÷ 4)
   = 20
   = 14h

   Program the PIP+ Line Address Offset registers.
   REG[80h]=14h
   REG[81h]=00h.

5. Enable the PIP+ window.

   Program the PIP+ Window Enable bit.
   REG[71h] bit 4 is set to 1.

8.3.4 SwivelView 270°

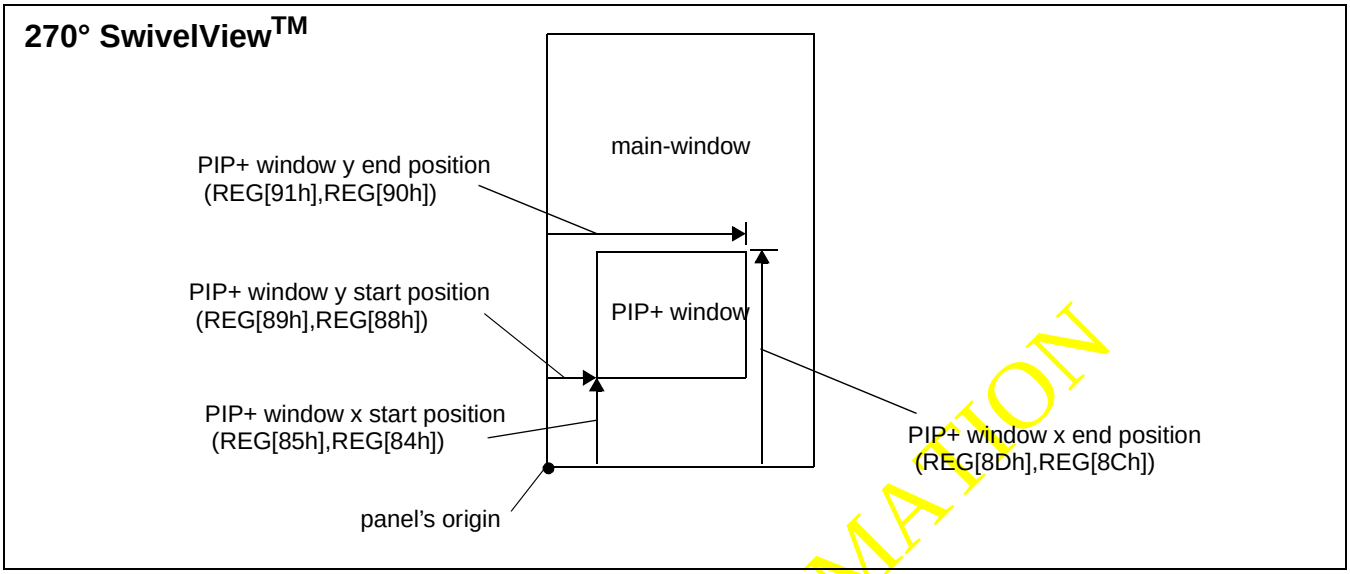


Figure 8-5: Picture-in-Picture Plus with SwivelView 270° enabled

SwivelView 270° is a mode in which both the main and PIP+ windows are rotated 270° counter-clockwise when shown on the panel. The images for each window are typically placed consecutively, with the main window image starting at address 0 and followed by the PIP+ window image. In addition, both images must start at addresses which are dword-aligned (the last two bits of the starting address must be 0).

Note

It is possible to use the same image for both the main window and PIP+ window. To do so, set the PIP+ Line Address Offset register (REG[81h],REG[80h]) to the same value as the Main Window Line Address Offset register (REG[79h],REG[78h]).

Example 8: In SwivelView 270°, program the PIP+ window registers for a 320x240 panel at 4 bpp, with the PIP+ window positioned at SwivelView 270° co-ordinates (60, 80) with a width of 120 and a height of 160.

1. Determine the value for the PIP+ Window X Positions and PIP+ Window Y Positions registers. Let the top left corner of the PIP+ window be (x1, y1), and let the bottom right corner be (x2, y2), where x2 = x1 + width - 1 and y2 = y1 + height - 1. The PIP+ Window X Positions register sets the vertical coordinates of the PIP+ window's top right and bottom left corner. The PIP+ Window Y Positions register sets the horizontal coordinates of the PIP+ window's top right and bottom left corner.

   The required values are calculated as follows:

   X Start Position
   = panel width - y2 - 1
   = 320 - (80 + 160 - 1) - 1
   = 80
   = 50h

   Y Start Position
   = x1 ÷ (32 ÷ bpp)
   = 60 ÷ (32 ÷ 4)
   = 7.5
   = 07h (truncated fractional part)

   X End Position
   = panel width - y1 - 1
   = 320 - 80 - 1
   = 239
   = EFh

   Y End Position
   = x2 ÷ (32 ÷ bpp)
   = (60 + 120 - 1) ÷ (32 ÷ 4)
   = 22.375
   = 16h (truncated fractional part)

2. Program the PIP+ Window X Positions register with the X Start Position in (REG[85h],REG[84h]) and the X End Position in (REG[8Dh],REG[8Ch]).
   REG[84h]=50h
   REG[85h]=00h
   REG[8Ch]=EFh
   REG[8Dh]=00h
   Program the PIP+ Window Y Positions register with the Y Start Position in (REG[89h],REG[88h]) and the Y End Position in (REG[91h],REG[90h]).
   REG[88h]=07h
   REG[89h]=00h
   REG[90h]=16h
   REG[91h]=00h

Note that the values of REG[84h] through REG[91h] do not go into effect until after REG[91h] is written.

Due to truncation, the dimensions of the PIP+ window may have changed. Recalculate the PIP+ window width and height below:
PIP Width
= ((REG[91h],REG[90h]) - (REG[89h],REG[88h]) + 1) × (32 ÷ bpp)
= (16h - 07h + 1) × (32 ÷ 4)
= 128 pixels **(note that this is different from the desired width)**

PIP+ Height
= (REG[8Dh],REG[8Ch]) - (REG[85h],REG[84h]) + 1
= EFh - 50h + 1
= 160 lines

3. Determine the PIP+ display start address.
The main window image must take up 320 x 240 pixels × bpp ÷ 8 = 9600h bytes. If the main window starts at address 0h, then the PIP+ window can start at 9600h.

PIP+ Stride
= image width × bpp ÷ 8
= 128 × 4 ÷ 8
= 64
= 40h

PIP+ display start address
= (desired byte address + ((PIP+ height - 1) × PIP+ Stride)) ÷ 4
= (9600h + ((160 - 1) × 64)) ÷ 4
= 12144
= 2F70h

Program the PIP+ Display Start Address registers.
REG[7Ch]=70h
REG[7Dh]=2Fh
REG[7Eh]=00h

4. Determine the PIP+ line address offset.

   number of dwords per line
   = image width ÷ (32 ÷ bpp)
   = 128 ÷ (32 ÷ 4)
   = 16
   = 10h

   Program the PIP+ Line Address Offset registers.
   REG[80h]=10h
   REG[81h]=00h.

5. Enable the PIP+ window.

   Program the PIP+ Window Enable bit.
   REG[71h] bit 4 is set to 1.

8.4 Limitations

8.4.1 SwivelView 0° and 180°

The PIP+ Line Address Offset registers (REG[81h],REG[80h]) requires the PIP+ window image *width* to be a multiple of 32 ÷ bits-per-pixel. If this formula is not satisfied, then the PIP+ Line Address Offset register must be programmed to the next larger value that satisfies the formula.

8.4.2 SwivelView 90° and 270°

The PIP+ Line Address Offset register (REG[81h],REG[80h]) requires the PIP+ window image *width* to be a multiple of 32 ÷ bits-per-pixel. If this formula is not satisfied, then the PIP+ Line Address Offset register must be programmed to the next larger value that satisfies the formula.

9 Hardware Abstraction Layer

9.1 Introduction

The S1D13708 Hardware Abstraction Layer (HAL) is a collection of routines intended to simplify the programming for the S5U13708B00B evaluation board. Programmers can use the HAL to assist in rapid software prototyping for the S5U13708B00B evaluation board.

The HAL routines are divided into discrete functional blocks. The functions for startup and clock control offer specific support for the S5U13708B00B evaluation board, while other routines demonstrate memory and register access techniques. For a complete list, see Table 9-1:, “HAL Library API” .

9.2 API for the HAL Library

The following table lists the functions provided by the S1D13708 HAL library.

Table 9-1: HAL Library API

Function	Description
Startup	
halAcquireController	This routine loads the driver required to access the S1D13708, locates the and returns the address of the controller.
halInitController	Initializes the controller for use. This includes setting the programmable clock and initializing registers as well as setting the lookup table and clearing video memory.
Memory Access	
halReadDisplay8	Reads one byte from display memory
halReadDisplay16	Reads one word from display memory
halReadDisplay32	Reads one double word from display memory
halWriteDisplay8	Writes one byte to display memory
halWriteDisplay16	Writes on word to display memory
halWriteDisplay32	Writes on double word to display memory
Register Access	
halReadReg8	Reads one byte from a control register
halReadReg16	Reads one word from a control register
halReadReg32	Reads one dword from a control register
halWriteReg8	Writes one byte to a control register
halWriteReg16	Writes one word to a control register
halWriteReg32	Writes one dword to a control registers
Clock Support	
halSetClock	Programs the ICD2061A Programmable Clock Generator.
halGetClock	Returns the frequency of the requested ICD2061A clock
Miscellaneous	
halGetVersionInfo	Returns a standardized startup banner message
halGetLastError	Returns the numerical value of the last error and optionally an ASCII string describing the error
halInitLUT	This routine sets the LUT to uniform values for color/mono panels at all color depths

9.2.1 Startup Routines

There are two routines dedicated to startup and initializing the S1D13708. Typically these two functions are the first two HAL routines a program will call.

The startup routine locates the S1D13708 controller and initializes HAL data structures. As the name suggests, the initialization routine prepares the S1D13708 for use. Splitting the startup functionality allows programs to start and locate the S1D13708 but delay or possibly never initialize the controller.

Boolean halAcquireController(UInt32 * pMem, UInt32 * pReg)

Description: This routine initializes data structures and initiates the link between the application software and the hardware. When the S1D13708 HAL is used this routine must be the first HAL function called.

The display buffer and register addresses, configured by 13708CFG, are used by halAcquireController(). When both addresses are set to 0, the software assumes an Intel platform, and will attempt to load the S1D13xxx driver. If this driver loads successfully, a check is then made for the S1D13708 evaluation board. If this board exists, halAcquireController() allocates display buffer and register addresses which are then returned via pMem and pReg.

If either the register or display buffer addresses in 13708CFG are non-zero, halAcquireController() will return these addresses via pMem and pReg.

Parameters:

pMem	Pointer to an unsigned 32-bit integer which will receive the offset to the first byte of display buffer memory. The offset may be cast to a pointer to access display memory.
pReg	Pointer to an unsigned 32-bit integer which will receive the offset to the first byte of register space. The offset may be cast to a pointer and to access S1D13708 registers. On Win32 systems the returned offsets correspond to a linear addresses within the caller's address space.

Return Value:

TRUE	(non-zero) The addresses in pMem and pReg are available for use. pMem will contain the offset to the first byte of display buffer memory. pRegs will contain the address of the first S1D13708 control register.
FALSE	(zero) The addresses in pMem and pReg cannot be used. If additional error information is required call halGetLastError().

Note

1. This routine **must** be called before any other HAL routine is called.
2. For programs written for the S1D13708 evaluation board, an application may call this routine to obtain pointers to the registers and display memory and then perform all S1D13708 accesses directly.
3. This routine does not modify S1D13708 registers or display buffer memory.

Boolean halInitController(UInt32 Flags)

Description: This routine performs the initialization portion of the startup sequence. Initialization of the S1D13708 evaluation board consists of several steps:
- Program the ICD2061A clock generator
- Set the initial state of the control
- Set the LUT to its default value
- Clear video memory

All display memory and nearly every control register can or will be affected by the initialization.

Any, or all, of the initialization steps may be bypassed according to values contained in the Flags parameter. This allows for conditional run-time changes to the initialization.

Parameters: Flags contains initialization specific information. The default action of the HAL is to perform all initialization steps. Flags contain specific instructions for bypassing certain initialization steps. Flags can be combined with the '|' symbol, such as fDONT_SET_CLOCKS | fDONT_INIT_REGS. The values for Flags are:

fDONT_SET_CLOCKS
Setting this flag causes initialization to skip programming the ICD2061A clock generator. Normally the clock on the S5U13708B00B is programmed to configured values during initialization.

fDONT_INIT_REGS
Bypass register initialization. Normally the init process sets the register values to a known state. Setting this flag bypasses this step.

fDONT_INIT_LUT
Bypass look-up table initialization.

fDONT_CLEAR_MEM
The final step of the initialization process is to clear display buffer memory. Setting this flag will bypass this step.

Return Value: TRUE (non-zero) if the initialization was successful.

FALSE (zero) if the HAL was unable to initialize the S1D13708
If additional error information is required call halGetLastError()

9.2.2 Memory Access

The S1D13708 HAL includes six memory access functions. The primary purpose of the memory access functions is to demonstrate how to access display memory using the C programming language. Most programs that need to access memory will bypass the HAL and instead use memory pointers.

Note

There are two means of accessing display memory: pointers into display memory, or through the indirect interface. The following memory access functions will support both types of access. However, the programmer must update the indirect interface functions in the HAL library to support the given hardware implementation. These indirect interface functions are all found in the file indirect.c in the HAL source code.

UInt8 halReadDisplay8(UInt32 Offset)

Description: Reads and returns the value of one byte of display memory.

Parameters: Offset — A 32 bit offset to the byte to be read from display memory.

Return Value: The value of the byte at the requested offset.

UInt16 halReadDisplay16(UInt32 Offset)

Description: Reads and returns the value of one word of display memory.

Parameters: Offset — A 32 bit byte offset to the word to be read from display memory. To prevent system slowdowns and possibly memory faults, Offset should be a word multiple.

Return Value: The value of the word at the requested offset.

UInt32 halReadDisplay32(UInt32 Offset)

Description: Reads and returns the value of one dword of display memory.

Parameters: Offset — A 32 bit byte offset to the dword to be read from display memory. To prevent system slowdowns and possibly memory faults, Offset should be a dword multiple.

Return Value: The value of the dword at the requested offset.

void halWriteDisplay8(UInt32 Offset, UInt8 Value, UInt32 Count)

Description: Writes a byte into display memory at the requested address.

Parameters: Offset — A 32 bit byte offset to the byte to be written to display memory.

Value — The byte value to be written to display memory.

Count — The number of times to repeat Value in memory. By including a count (or loop) value this function can efficiently fill display memory.

Return Value: Nothing.

void halWriteDisplay16(UInt32 Offset, UInt16 Value, UInt32 Count)

Description: Writes a word into display memory at the requested offset.

Parameters:

Offset	a 32 bit byte offset to the byte to be written to display memory. To prevent system slowdowns and possibly memory faults, Offset should be a word multiple.
Value	the word value to be written to display memory.
Count	the number of times to repeat the Value in memory. By including a count (or loop) value this function can efficiently fill display memory.

Return Value: Nothing.

void halWriteDisplay32(UInt32 Offset, UInt32 Value, UInt32 Count)

Description: Writes a dword into display memory at the requested offset.

Parameters:

Offset	A 32 bit byte offset to the byte to be written to display memory. To prevent system slowdowns and possibly memory faults, Offset should be a dword multiple.
Value	The dword value to be written to display memory.
Count	The number of times to repeat the Value in memory. By including a count (or loop) value this function can efficiently fill display memory.

Return Value: Nothing.

9.2.3 Register Access

The S1D13708 HAL includes six register access functions. The primary purpose of the register access functions is to demonstrate how to access the S1D13708 control registers using the C programming language. Most programs that need to access the registers will bypass the HAL and instead use register pointers.

There are some register values which must be combined, such as the Main Window Line Address Offset in REG[78h] and REG[79h]. Software must take into account whether the given platform is big or little endian when combining such register values. To guarantee that the combined register values are correct, do the following in C:

Read word from reg[index]	val16 = (halReadReg8(index+1) << 8) | halReadReg8(index);
Read dword from reg[index]	val32 =(halReadReg8(index+3) << 24) | (halReadReg8(index+2) << 16) | (halReadReg8(index+1) << 8) halReadReg8(index);
Write word (val16) to reg[index]	halWriteReg8(index, (val16 & 0xff)); halWriteReg8(index+1, (val16 >> 8) & 0xff);

Write dword (val32) to reg[index]	halWriteReg8(index, (val32 & 0xff));
	halWriteReg8(index+1, (val32 >> 8) & 0xff);
	halWriteReg8(index+2, (val32 >> 16) & 0xff);
	halWriteReg8(index+3, (val32 >> 24) & 0xff);

Note

There are two means of accessing registers: pointers into register addressing space, or through the indirect interface. The following register access functions will support both types of access. However, the programmer must update the indirect interface functions in the HAL library to support the given hardware implementation. These indirect interface functions are all found in the file indirect.c in the HAL source code.

UInt8 halReadReg8(UInt32 Index)

Description: Reads and returns the contents of one byte of an S1D13708 register at the requested offset. No S1D13708 registers are changed.

Parameters:

Index	32 bit offset to the register to read. Index is zero-based from the beginning of register address space.

Return Value: The value read from the register.

UInt16 halReadReg16(UInt32 Index)

Description: Reads and returns the contents of one word of an S1D13708 register at the requested offset. No S1D13708 register are changed.

Parameters:

Index	32 bit offset to the register to read. Index is zero-based from the beginning of register address space.

Return Value: The word value read from the register.

UInt16 halReadReg32(UInt32 Index)

Description: Reads and returns the dword value of an S1D13708 register at the requested offset. No S1D13708 register are changed.

Parameters:

Index	32 bit offset to the register to read. Index is zero-based from the beginning of register address space.

Return Value: The dword value read from the register.

void halWriteReg8(UInt32 Index, UInt8 Value)

Description: Writes an 8 bit value to the register at the requested offset.

Parameters:

Index	32 bit offset to the register to write. Index is zero-based from the beginning of register address space.
Value	The byte value to write to the register.

Return Value: Nothing.

void halWriteReg16(UInt32 Index, UInt16 Value)

Description: Writes a 16 bit value to the S1D13708 register at the requested offset.

Parameters:

Index	32 bit byte offset to the register to write. Index is zero-based from the beginning of register address space.
Value	The word value to write to the register.

Return Value: Nothing.

void halWriteReg32(UInt32 Index, UInt32 Value)

Description: Writes a 32 bit value (dword) to the register at the requested offset.

Parameters:

Index	32 bit byte offset to the register to write. Index is zero-based from the beginning of register address space.
Value	The dword value to write to the register.

Return Value: Nothing.

9.2.4 Clock Support

To maximize flexibility, S5U13708B00B evaluation boards include a programmable clock. The following HAL routines provide support for the programmable clock.

Boolean halSetClock(UInt32 ClkiFreq, UInt32 Clki2Freq)

Description: This routine program the ICD2061A programmable clock generator to the specified frequency.

Parameters:

ClkiFreq	The desired frequency, in Hz, for CLKI.
Clki2Freq	The desired frequency, in Hz, for CLKI2.

Return Value: TRUE (non-zero) if the function was successful in setting the clock.

FALSE (zero) if there was an error detected while trying to set the clock.
If additional error information is required call halGetLastError().

UInt32 halGetClock(CLOCKSELECT Clock)

Description: Returns the frequency of the clock input identified by 'Clock'.

Parameters: Clock Indicates which clock to read. This value can be CLKI or CLKI2.

Return Value: The frequency, in Hz, of the requested clock.

9.2.5 Miscellaneous

The miscellaneous function are an assortment of routines, determined to be beneficial to a number of programs and hence warranted being included in the HAL.

void halGetVersionInfo(const char * szProgName, const char * szDesc, const char * szVersion, char * szRetStr, int Length)

Description: This routine creates a standardized startup banner by merging program and HAL specific information. The newly formulated string is returned to the calling program for display.

The final formatted string will resemble:

13708PROGRAM - Internal test and diagnostic program - Build 1234 [HAL: 1234]
Copyright (c) 2001 Epson Research and Development, Inc.
All rights reserved.

Parameters:

szProgName Pointer to an ASCIIZ string containing the name of the program. (e.g. “PROGRAM”).

szDesc Pointer to an ASCIIZ string containing a description of what this program is intended to do (e.g. “Internal test and diagnostic program”).

szVersion Pointer to an ASCIIZ string containing the build info for this program. This should be the revision info string as updated by Microsoft Visual Source Safe© (e.g. “$Revision: 30 $”).

szRetStr Pointer to a buffer into which the product and version information will be formatted into.

Length Total number of bytes in the string pointed to by szRetStr. This function will write Length or fewer bytes to the buffer pointed to by szRetStr.

Return Value: Nothing.

int halGetLastError(char * ErrMsg, int MaxSize)

Description: This routine retrieves the last error detected by the HAL.

Parameters: ErrMsg When halGetLastError() returns, ErrMsg will point to the textual error message. If ErrMsg is NULL then only the error code will be returned.

MaxSize Maximum number of bytes, including the final '\0', that can be placed in the string pointed to by ErrMsg.

Return Value: The numerical value of the internal error number.

HALEXTERN void halInitLUT(void)

Description: To standardize the appearance of test and validation programs, it was decided the HAL would have the ability to set the lookup table to uniform values.

The routine cracks the color depth and display type to determine which LUT values to use and proceeds to write the LUT entries.

Parameters: None

Return Value: Nothing.

10 Sample Code

Example source code demonstrating programming the S1D13708 using the HAL library is available on the internet at www.erd.epson.com.

11 Sales and Technical Support

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

THIS PAGE LEFT BLANK

READ-ONLY CONFIGURATION REGISTERS

REG[00h] Revision Code Register [1] — RO

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Product Code = 001101 (0Dh)						Revision Code = 00	

REG[01h] Display Buffer Size Register — RO

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Display Buffer Size = 00010100 (14h)							

REG[02h] Configuration Readback Register — RO

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CNF7 Status	CNF6 Status	CNF5 Status	CNF4 Status	CNF3 Status	CNF2 Status	CNF1 Status	CNF0 Status

CLOCK CONFIGURATION REGISTERS

REG[04h] Memory Clock Configuration Register [2] — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a		MCLK Divide Select bits 1-0		n/a			

REG[05h] Pixel Clock Configuration Register [3,4] — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	PCLK Divide Select bits 2-0			n/a		PCLK Source Select bits 1-0	

LOOK-UP TABLE REGISTERS

REG[08h] Look-Up Table Blue Write Data Register — WO

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LUT Blue Write Data bits 5-0						n/a	

REG[09h] Look-Up Table Green Write Data Register — WO

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LUT Green Write Data bits 5-0						n/a	

REG[0Ah] Look-Up Table Red Write Data Register — WO

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LUT Red Write Data bits 5-0						n/a	

REG[0Bh] Look-Up Table Write Address Register — WO

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LUT Write Address bits 7-0							

REG[0Ch] Look-Up Table Blue Read Data Register — RO

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LUT Blue Read Data bits 5-0						n/a	

REG[0Dh] Look-Up Table Green Read Data Register — RO

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LUT Green Read Data bits 5-0						n/a	

REG[0Eh] Look-Up Table Red Read Data Register — RO

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LUT Red Write Data bits 5-0						n/a	

REG[0Fh] Look-Up Table Read Address Register — WO

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LUT Read Address bits 7-0							

PANEL CONFIGURATION REGISTERS

REG[10h] Panel Type Register [5,6] — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Panel Data Format Select	Color/Mono Panel Select	Panel Data Width bits 1-0		Active Panel Res. Select	n/a	Panel Type bits 1-0	

REG[11h] MOD Rate Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a		MOD Rate bits 5-0					

REG[12h] Horizontal Total Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Horizontal Total bits 6-0						

REG[14h] Horizontal Display Period Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Horizontal Display Period bits 6-0						

REG[16h] Horizontal Display Period Start Position Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Horizontal Display Period Start Position bits 7-0							

REG[17h] Horizontal Display Period Start Position Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						Horizontal Display Period Start Position bits 9-8	

REG[18h] Vertical Total Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Vertical Total bits 7-0							

REG[19h] Vertical Total Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						Vertical Total bits 9-8	

REG[1Ch] Vertical Display Period Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Vertical Display Period bits 7-0							

REG[1Dh] Vertical Display Period Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						Vertical Display Period bits 9-8	

REG[1Eh] Vertical Display Period Start Position Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Vertical Display Period Start Position bits 7-0							

REG[1Fh] Vertical Display Period Start Position Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						Vertical Display Period Start Position bits 9-8	

REG[20h] FPLINE Pulse Width Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FPLINE Pulse Polarity	FPLINE Pulse Width bits 6-0						

REG[22h] FPLINE Pulse Start Position Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FPLINE Pulse Start Position bits 7-0							

REG[23h] FPLINE Pulse Start Position Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						FPLINE Pulse Start Position bits 9-8	

REG[24h] FPFRAME Pulse Width Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FPFRAME Pulse Polarity	n/a				FPFRAME Pulse Width bits 2-0		

REG[26h] FPFRAME Pulse Start Position Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FPFRAME Pulse Start Position bits 7-0							

REG[27h] FPFRAME Pulse Start Position Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						FPFRAME Pulse Start Position bits 9-8	

REG[28h] D-TFD GCP Index Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a			D-TFD GCP Index bits 4-0				

REG[2Ch] D-TFD GCP Data Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D-TFD GCP Data bits 7-0							

DISPLAY MODE REGISTERS

REG[70h] Display Mode Register [7] — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Display Blank	Dithering Disable	HW Video Invert Enable	SW Video Invert	n/a	Bit-per-pixel Select bits 2-0		

REG[71h] Special Effects Register [8] — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Display Data Word Swap	Display Data Byte Swap	n/a	PIP+ Window Enable	n/a		SwivelView™ Mode Select bits 1-0	

REG[74h] Main Window Display Start Address Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Main Window Display Start Address bits 7-0							

REG[75h] Main Window Display Start Address Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Main Window Display Start Address bits 15-8							

REG[76h] Main Window Display Start Address Register 2 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a							Main Window Display Start Address Bit 16

REG[78h] Main Window Line Address Offset Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Main Window Line Address Offset bits 7-0							

REG[79h] Main Window Line Address Offset Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						Main Window Line Address Offset bits 9-8	

PICTURE-IN-PICTURE PLUS REGISTERS

REG[7Ch] PIP+ Window Display Start Address Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIP+ Window Display Start Address bits 7-0							

REG[7Dh] PIP+ Window Display Start Address Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIP+ Window Display Start Address bits 15-8							

REG[7Eh] PIP+ Window Display Start Address Register 2 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a							PIP+ Window Display Start Address Bit 16

REG[80h] PIP+ Window Line Address Offset Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIP+ Window Line Address Offset bits 7-0							

REG[81h] PIP+ Window Line Address Offset Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						PIP+ Window Line Address Offset bits 9-8	

REG[84h] PIP+ Window X Start Position Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIP+ Window X Start Position bits 7-0							

REG[85h] PIP+ Window X Start Position Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						PIP+ Window X Start Position bits 9-8	

REG[88h] PIP+ Window Y Start Position Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIP+ Window Y Start Position bits 7-0							

REG[89h] PIP+ Window Y Start Position Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						PIP+ Window Y Start Position bits 9-8	

REG[8Ch] PIP+ Window X End Position Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIP+ Window X End Position bits 7-0							

REG[8Dh] PIP+ Window X End Position Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						PIP+ Window X End Position bits 9-8	

REG[90h] PIP+ Window Y End Position Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PIP+ Window Y End Position bits 7-0							

REG[91h] PIP+ Window Y End Position Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						PIP+ Window Y End Position bits 9-8	

MISCELLANEOUS REGISTERS

REG[A0h] Power Save Configuration Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VNDP Status (RO)	n/a			Mem Contr PS Status (RO)	n/a		PS Mode Enable

REG[A1h] Reserved — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a							Reserved

REG[A2h] Software Reset Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	n/a						Software Reset (WO)

REG[A3h] Reserved — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	n/a						

REG[A4h] Scratch Pad Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Scratch Pad bits 7-0							

REG[A5h] Scratch Pad Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Scratch Pad 15-8							

GENERAL IO PINS REGISTERS

REG[A8h] General Purpose IO Pins Configuration Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	GPIO6 Pin IO Config	GPIO5 Pin IO Config	GPIO4 Pin IO Config	GPIO3 Pin IO Config	GPIO2 Pin IO Config	GPIO1 Pin IO Config	GPIO0 Pin IO Config

REG[A9h] General Purpose IO Pins Configuration Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GPIO Pin Input Enable	Reserved						

REG[AAh] General Purpose IO Pins Status/Control Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	GPIO6 Pin IO Status	GPIO5 Pin IO Status	GPIO4 Pin IO Status	GPIO3 Pin IO Status	GPIO2 Pin IO Status	GPIO1 Pin IO Status	GPIO0 Pin IO Status

REG[ADh] General Purpose IO Pins Status/Control Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GPO Control	Reserved						

PWM CLOCK AND CV PULSE CONFIGURATION REGISTERS

REG[B0h] PWM Clock / CV Pulse Control Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM Clock Force High	n/a		PWM Clock Enable	CV Pulse Force High	CV Pulse Burst Status (RO)	CV Pulse Burst Start	CV Pulse Enable

REG[B1h] PWM Clock / CV Pulse Configuration Register [9,10] — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM Clock Divide Select bits 3-0				CV Pulse Divide Select bits 2-0			PWMCLK Source Select

REG[B2h] CV Pulse Burst Length Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CV Pulse Burst Length bits 7-0							

REG[B3h] PWMOUT Duty Cycle Register [11] — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMOUT Duty Cycle bits 7-0							

EXTENDED REGISTERS

REG[C0h] Memory Access Pointer 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Memory Access Pointer bits 7-0							

REG[C1h] Memory Access Pointer 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Memory Access Pointer bits 15-8							

REG[C2h] Memory Access Pointer 2 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a							Memory Access Pointer bit 16

REG[C4h] Memory Access Start — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a							

REG[C5h] Extended Panel Type Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						Extended Panel Type bits 1-0	

REG[C6h] Memory Access Select Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a							Memory Access Select

REG[C7h] Ink Transparent Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Ink Transparent bits 7-0							

REG[C8h] Ink Transparent Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Ink Transparent bits 15-8							

REG[C9h] Ink Layer Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
							Ink Layer Enable

REG[CAh] OSCP Configuration Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	OSCP Enable bits 1-0			n/a			BCLK Source Select

REG[CBh] TFT Data Compare Invert Enable Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a							Data Compare Invert Enable

REG[D0h] TFT Type 2 VCLK Configuration Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a			VCLK Hold bits 1-0		n/a	VCLK Setup bits 1-0	

REG[D1h] TFT Type 2 AP Configuration Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
POL Type	n/a	AP Pulse Width bits 2-0			n/a	AP Rising Position bits 1-0	

REG[D4h] TFT Type 3 Control Signal Enable Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDME Control	XSTBY Control	XOHV Control	XRESV Control	XRESH Control	PCLK2 Enable	PCLK1 Enable	n/a

REG[D5h] TFT Type 3 OE Rising Edge Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OE Rising Edge Position bits 7-0							

REG[D6h] TFT Type 3 OE Pulse Width Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OE Pulse Width bits 7-0							

REG[D7h] TFT Type 3 POL Toggle Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
POL Toggle Position bits 7-0							

REG[D8h] TFT Type 3 VCOM Toggle Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VCOM Toggle Position bits 7-0							

REG[D9h] TFT Type 3 CPV Pulse Width Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPV Pulse Width bits 7-0							

REG[DAh] TFT Type 3 XOEV Rising Edge Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XOEV Rising Edge Position bits 7-0							

REG[DBh] TFT Type 3 XOEV GPIO5 Falling Edge Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XOEV Falling Edge Position bits 7-0							

REG[DCh] TFT Type 3 PCLK Divide Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a		PCLK2 Divide Rate bits 1-0		PCLK1 Divide Rate bits 3-0			

REG[E0h] TFT Type 3 Partial Mode Display Area Control Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a			Partial Mode Display Enable	Parial Mode Display Type Select	Area 2 Display Enable	Area 1 Display Enable	Area 0 Display Enable

REG[E1h] TFT Type 3 Partial Mode Display Refresh Cycle Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Mode Display Refresh Cycle bits 5-0						

REG[E2h] TFT Type 3 Partial Area 0 X Start Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Area 0 X Start Position bits 5-0						

REG[E3h] TFT Type 3 Partial Area 0 Y Start Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Area 0 Y Start Position bits 5-0						

REG[E4h] TFT Type 3 Partial Area 0 X End Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Area 0 X End Position bits 5-0						

REG[E5h] TFT Type 3 Partial Area 0 Y End Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Area 0 Y End Position bits 5-0						

REG[E6h] TFT Type 3 Partial Area 1 X Start Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Area 1 X Start Position bits 5-0						

REG[E7h] TFT Type 3 Partial Area 1 Y Start Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Area 1 Y Start Position bits 5-0						

REG[E8h] TFT Type 3 Partial Area 1 X End Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Area 1 X End Position bits 5-0						

REG[E9h] TFT Type 3 Partial Area 1 Y End Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Area 1 Y End Position bits 5-0						

REG[EAh] TFT Type 3 Partial Area 2 X Start Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Area 2 X Start Position bits 5-0						

REG[EBh] TFT Type 3 Partial Area 2 Y Start Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Area 2 Y Start Position bits 5-0						

REG[ECh] TFT Type 3 Partial Area 2 X End Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Area 2 X End Position bits 5-0						

REG[EDh] TFT Type 3 Partial Area 2 Y End Position Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a	Partial Area 2 Y End Position bits 5-0						

REG[F0h] TFT Type 3 Command 0 Store Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Command 0 Store bits 7-0							

REG[F1h] TFT Type 3 Command 0 Store Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a				Command 0 Store bits 11-8			

REG[F2h] TFT Type 3 Command 1 Store Register 0 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Command 1 Store bits 7-0							

REG[F3h] TFT Type 3 Command 1 Store Register 1 — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a				Command 1 Store bits 11-8			

REG[F4h] TFT Type 3 Command Send Request Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a							Command Send Request

REG[F5h] Source Driver IC Number Register — RW

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
n/a						Source Driver IC Number bits 1-0	

Notes

1 REG[00h] These bits are used to identify the SED13708. For the SED13708, the product code should be 001101.

2 REG[04h] Memory Clock Configuration Register

MCLK Divide Select Bits	BCLK to MCLK Frequency Ratio
00	1:1
01	2:1
10	3:1
11	4:1

3 REG[05h] Pixel Clock Configuration Register

PCLK Divide Select Bits	PCLK Source to PCLK Frequency Ratio
000	1:1
001	2:1
010	3:1
011	4:1
1XX	8:1

4 REG[05h] Pixel Clock Configuration Register

PCLK Source Select Bits	PCLK Source
00	MCLK
01	BCLK
10	CLKI
11	CLKI2

5 REG[10h] Panel Type Register

Panel Data Width Bits [1:0]	Passive LCD Panel Data Width Size	Active Panel Data Width Size
00	4-bit	9-bit
01	8-bit	12-bit
10	16-bit	18-bit
11	Reserved	Reserved

6 REG[10h] Panel Type Register

REG[10h] Bits[1:0]	Panel Type
00	STN
01	TFT
10	HR-TFT
11	D-TFD

7 REG[70h] Display Mode Register

Bit-per-pixel Select Bits [1:0]	Color Depth (bpp)	Maximum Number of Colors/Shades		Max. No. Of Simultaneously Displayed Colors/ Shades
		Passive Panel (Dithering On)	TFT Panel	
000	1 bpp	256K/64	256K/64	2/2
001	2 bpp	256K/64	256K/64	4/4
010	4 bpp	256K/64	256K/64	16/16
011	8 bpp	256K/64	256K/64	256/64
100	16 bpp	64K/64	64K/64	64K/64
101, 110, 111	Reserved	n/a	n/a	n/a

8 REG[71h] Special Effects Register

SwivelView™ Mode Select Bits	SwivelView™ Orientation
00	Normal
01	90°
10	180°
11	270°

9 REG[B1h] PWM Clock / CV Pulse Configuration Register

PWM Clock Divide Select Bits [3:0]	PWM Clock Divide Amount
0h	1
1h	2
2h	4
3h	8
...	...
Ch	4096
Dh-Fh	Reserved

10 REG[B1h] PWM Clock / CV Pulse Configuration Register

CV Pulse Divide Select Bits [2:0]	CV Pulse Divide Amount
0h	1
1h	2
2h	4
3h	8
...	...
7h	128

11 REG[B3h] PWMOUT Duty Cycle Register

PWMOUT Duty Cycle [7:0]	PWMOUT Duty Cycle
00h	Always Low
01h	High for 1 out of 256 clock periods
02h	High for 2 out of 256 clock periods
...	...
FFh	High for 255 out of 256 clock periods

EPSON®

S1D13708 Embedded Memory LCD Controller

13708CFG Configuration Program

Document Number: X39A-B-001-01

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

13708CFG

13708CFG is an interactive Windows® 9x/ME/NT/2000 program that calculates register values for a user-defined S1D13708 configuration. The configuration information can be used to directly alter the operating characteristics of the S1D13708 utilities or any program built with the Hardware Abstraction Layer (HAL) library. Alternatively, the configuration information can be saved in a variety of text file formats for use in other applications.

S1D13708 Supported Evaluation Platforms

13708CFG runs on PC system running Windows 9x/ME/NT/2000 and can modify Win32.exe files and .s9 format files.

Installation

Create a directory for **13708cfg.exe** and copy the files **13708cfg.exe** and **panels.def** to that directory. **Panels.def** contains configuration information for a number of panels and must reside in the same directory as **13708cfg.exe**.

Usage

To start 13708CFG from the Windows desktop, double click the program icon.

To start 13708CFG from a Windows command prompt, change to the directory **13708cfg.exe** was installed to and type the command **13708cfg**.

The basic procedure for using 13708CFG is:

1. Start 13708CFG as described above.
2. Open an existing file to serve as a starting reference point (this step is optional).
3. Modify the configuration. For specific information on editing the configuration, see “13708CFG Configuration Tabs” on page 6.
4. Save the new configuration. The configuration information can be saved in two ways; as an ASCII text file or by modifying the executable image on disk.

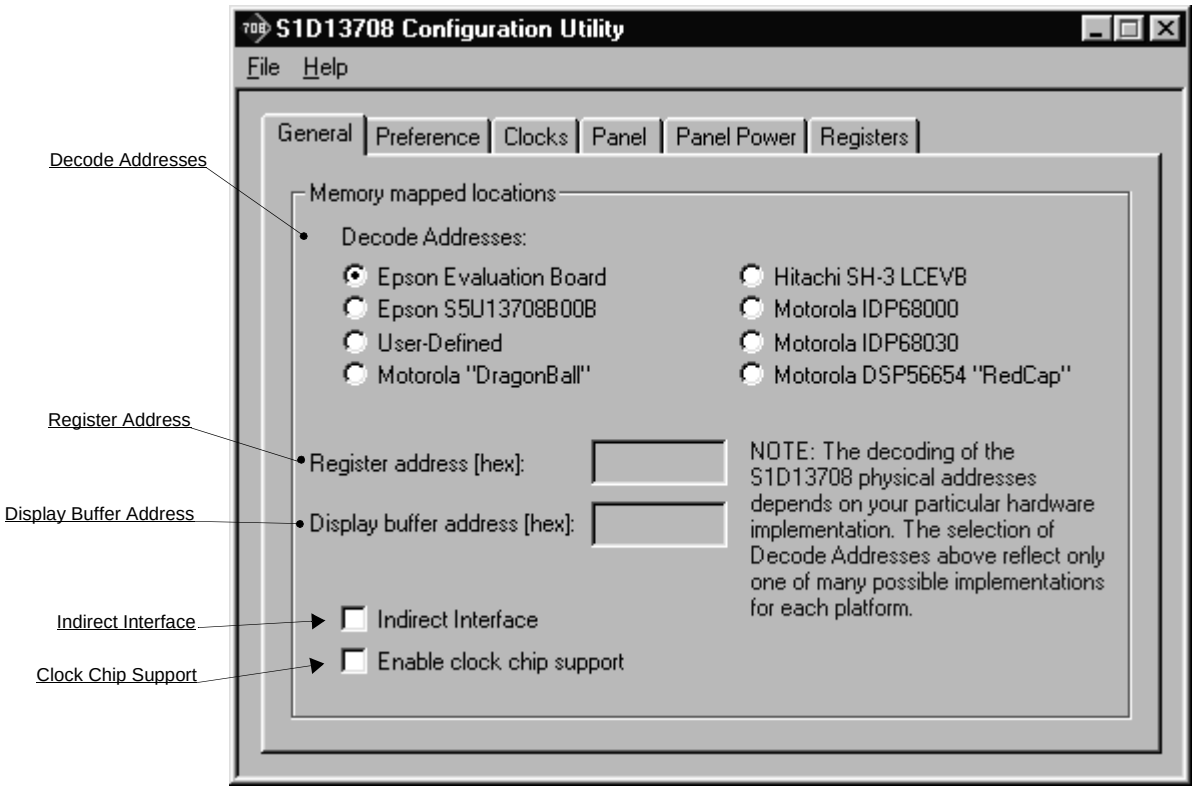
   Several ASCII text file formats are supported. Most are formatted C header files used to build display drivers or standalone applications.

   Utility files based on the Hardware Abstraction Layer (HAL) can be modified directly by 13708CFG.

13708CFG Configuration Tabs

13708CFG provides a series of tabs which can be selected at the top of the main window. Each tab allows the configuration of a specific aspect of S1D13708 operation. The following sections describe the purpose and use of each of the tabs.

General Tab



The General tab contains settings that define the S1D13708 operating environment.

Decode Addresses

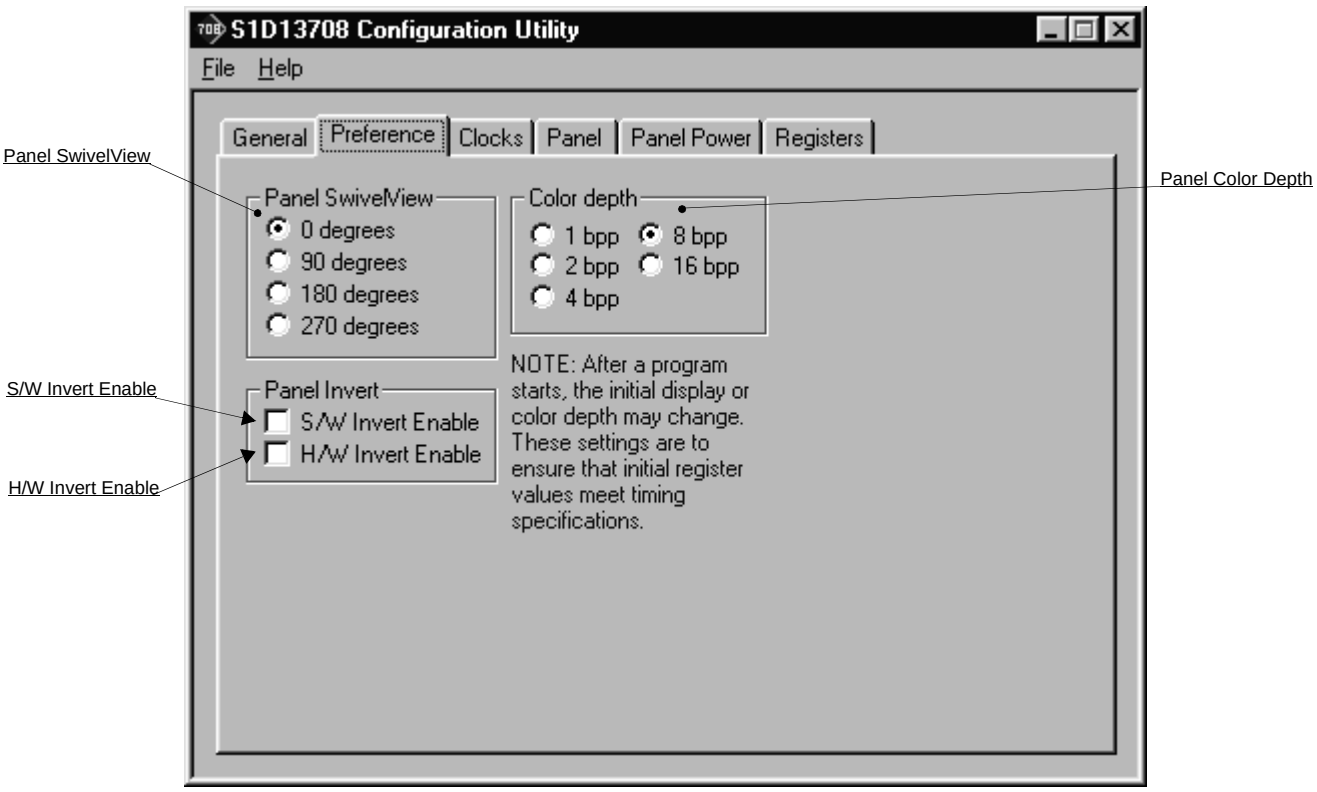
Selecting one of the listed evaluation platforms changes the values for the “Register address” and “Display buffer address” fields. The values used for each evaluation platform are examples of possible implementations as used by the Epson S1D13708 evaluation board. If your hardware implementation differs from the addresses used, select the User-Defined option and enter the correct addresses for “Register address” and “Display buffer address”.

Note

When “Epson S5U13708B00B” is selected, the register and display buffer addresses are blanked because the evaluation board uses the PCI interface and the decode addresses are determined by the system BIOS during boot-up.
When “Epson Evaluation Board” is selected, the register and display buffer addresses are blanked because the evaluation board communicates via the serial port.

Register Address	The physical address of the start of register decode space (in hexadecimal). This field is automatically set according to the Decode Address unless the “User-Defined” decode address is selected.
Display Buffer Address	The physical address of the start of display buffer decode space (in hexadecimal). This field is automatically set according to the Decode Address unless the “User-Defined” decode address is selected.
Indirect Interface	This setting selects software support for the S1D13708 Indirect Bus Interface.
Clock Chip Support	The S5U13708B00B evaluation board implements a Cypress ICD2061A Clock Synthesizer which can be used to generate CLKI and CLKI2. When this box is checked, GPIO[3:1] are reserved for Clock Synthesizer support. Selecting a HR-TFT panel will disable this feature as the HR-TFT requires GPIO[3:0]. **Note that this feature is only available when using the S5U13708B00B.**

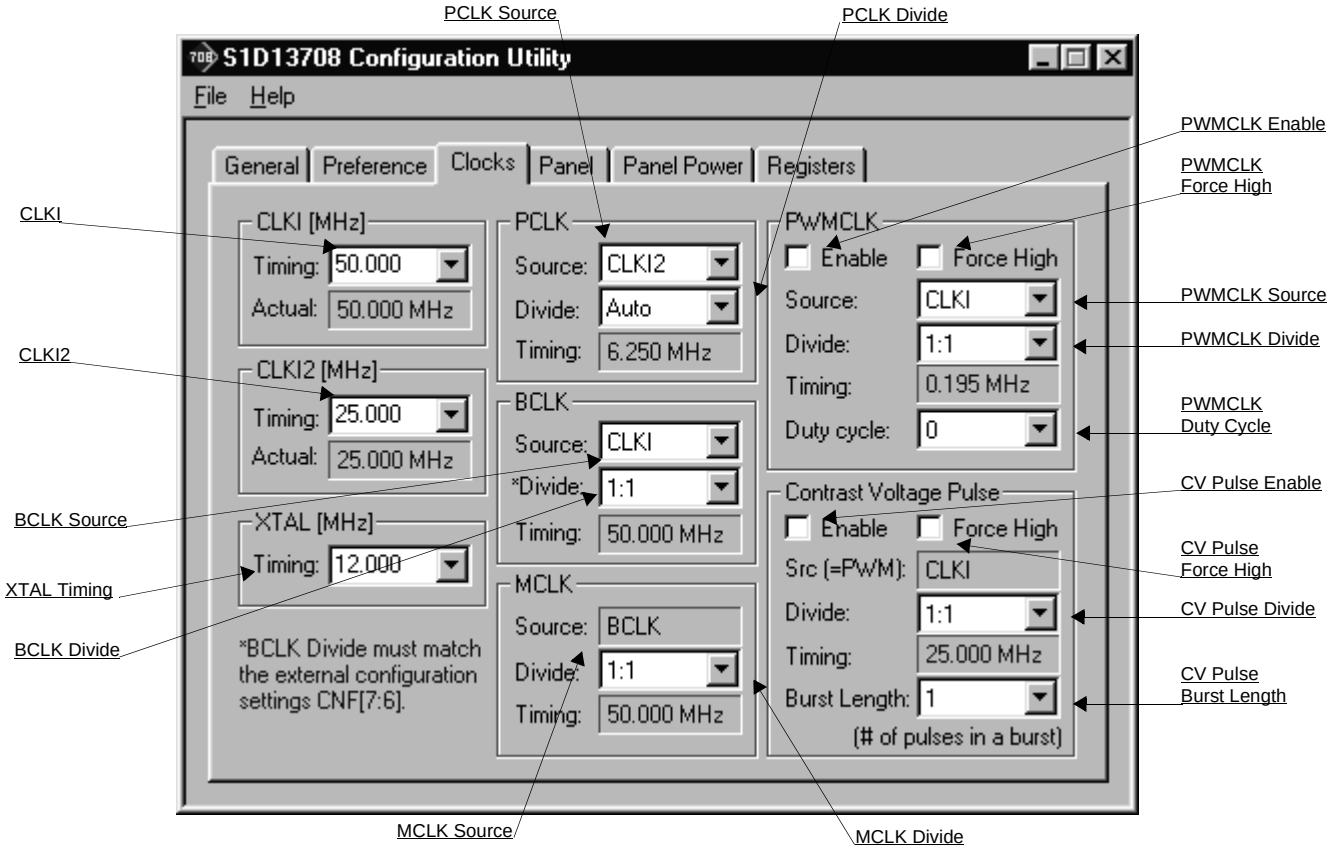
Preferences Tab



The Preference tab contains settings pertaining to the initial display state. During runtime these settings may be changed.

Panel SwivelView	The S1D13708 SwivelView feature is capable of rotating the image displayed 90°, 180°, or 270° in a counter-clockwise direction. This setting determines the initial orientation of the panel.
Panel Invert	The S1D13708 can invert the display data going to the LCD panel. The display data is inverted after the Look-Up Table, which means colors are truly inverted.
S/W Invert Enable	The Video Invert feature can be controlled by software using REG[70h] bit 4. When this box is checked, the Software Video Invert bit is set to one and video data is inverted. If the box is unchecked, the bit is set to zero and video data remains normal.
H/W Invert Enable	The Video Invert feature can be controlled by hardware if the GPIO0 is available. Hardware control is not possible if an HR-TFT or D-TFD panel is used as both panels use GPIO0 as an LCD control signal.
Color Depth	Sets the initial color depth on the LCD panel.

Clocks Tab



The Clocks tab simplifies the selection of input clock frequencies and the sources of internal clocking signals. For further information regarding clocking and clock sources, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

It is the responsibility of the system designer to ensure that the correct clock frequencies are supplied to the S1D13708.

Note

Changing clock values may modify or invalidate Panel settings. Confirm all settings on the Panel tab after modifying any clock settings.

The S1D13708 may use one or two clock sources. Two clock sources allow greater flexibility in display type and memory speed.

CLKI — This setting determines the frequency of CLKI.

- Timing — Set this value by selecting a preset frequency from the drop down list or entering the desired frequency (MHz) in the edit box.
- Actual — This field displays the actual value of the CLKI frequency. If "Enable clock chip support" is selected on the General Tab, then this value may differ slightly from the value entered in the timing control.

CLKI2 — This setting determines the frequency of CLKI2.

- Timing — Set this value by selecting a preset frequency from the drop down list or entering the desired frequency (MHz) in the edit box.
- Actual — This field displays the actual value of the CLKI2 frequency. If "Enable clock chip support" is selected on the General Tab, then this value may differ slightly from the value entered in the timing control.

XTAL — These settings select the clock frequency for the Crystal Oscillator Input (XTAL).

- Timing — This field selects the actual XTAL frequency used by the configuration process.

PCLK — These settings select the clock signal source and divisor for the pixel clock (PCLK).

- Source — Selects the PCLK source. Possible sources include CLKI, CLKI2, BCLK, MCLK or XTAL. Note that BCLK and MCLK may be previously divided from CLKI or CLKI2.
- Divide — Specifies the divide ratio for the clock source. The divide ratio is applied to the PCLK source to derive PCLK.

  Selecting "Auto" for the divisor allows the configuration program to calculate the best clock divisor. Unless a very specific clocking is being specified, it is best to leave this setting on "Auto".

Timing	This field shows the actual PCLK used by the configuration process.
BCLK	These settings select the clock source and divisor for the internal bus interface clock (BCLK).
Source	Selects the BCLK source. Possible sources include CLKI and XTAL. Note that XTAL can be selected only when the indirect interface is enabled.
Divide	Specifies the divide ratio for the clock source. The divide ratio is applied to the BCLK source to derive BCLK.
Timing	This field shows the actual BCLK frequency used by the configuration process.
MCLK	These settings select the clock source and input clock divisor for the internal memory clock (MCLK). For the best performance, MCLK should be set as close to the maximum (50 MHz) as possible.
Source	The MCLK source is always BCLK.
Divide	Specifies the divide ratio for the clock source. The divide ratio is applied to the MCLK source to derive MCLK. This divide ratio should be left at 1:1 unless the resultant MCLK is greater that 50MHz.
Timing	This field shows the actual MCLK frequency used by the configuration process.

PWMCLK

These controls configure various PWMCLK settings. The PWMCLK is the internal clock used by the Pulse Width Modulator for output to the panel.

Enable

When this box is checked, the PWMCLK circuitry is enabled.

Force High

The signal PWMOUT is forced high when this box is checked. When not checked, PWMOUT will be low if PWM is not enabled or will change state according to the configured values when PWM is enabled.

Source

Selects the PWMCLK source. Possible sources include CLKI, CLKI2 and XTAL. Note that CLKI2 is available for a PWMCLK source clock if PCLK does *not* use XTAL. Also note that XTAL is available for a PWMCLK source clock if PCLK does *not* use CLKI2.

Divide

Specifies the divide ratio for the clock source. The divide ratio is applied to the PWMCLK source to derive PWMCLK.

Note

After this divide is applied, PWMCLK is further divided by 256 to achieve the final PWMCLK frequency.

Timing

This field shows the actual PWMCLK frequency used by the configuration process.

Duty Cycle

Selects the number of cycles that PWMOUT is high out of 256 clock periods.

Contrast Voltage Pulse

These controls configure various Contrast Voltage (CV) Pulse settings. The CV Pulse is provided for panels which support the contrast voltage function.

Enable

When this box is checked, the CV Pulse circuitry is enabled.

Force High

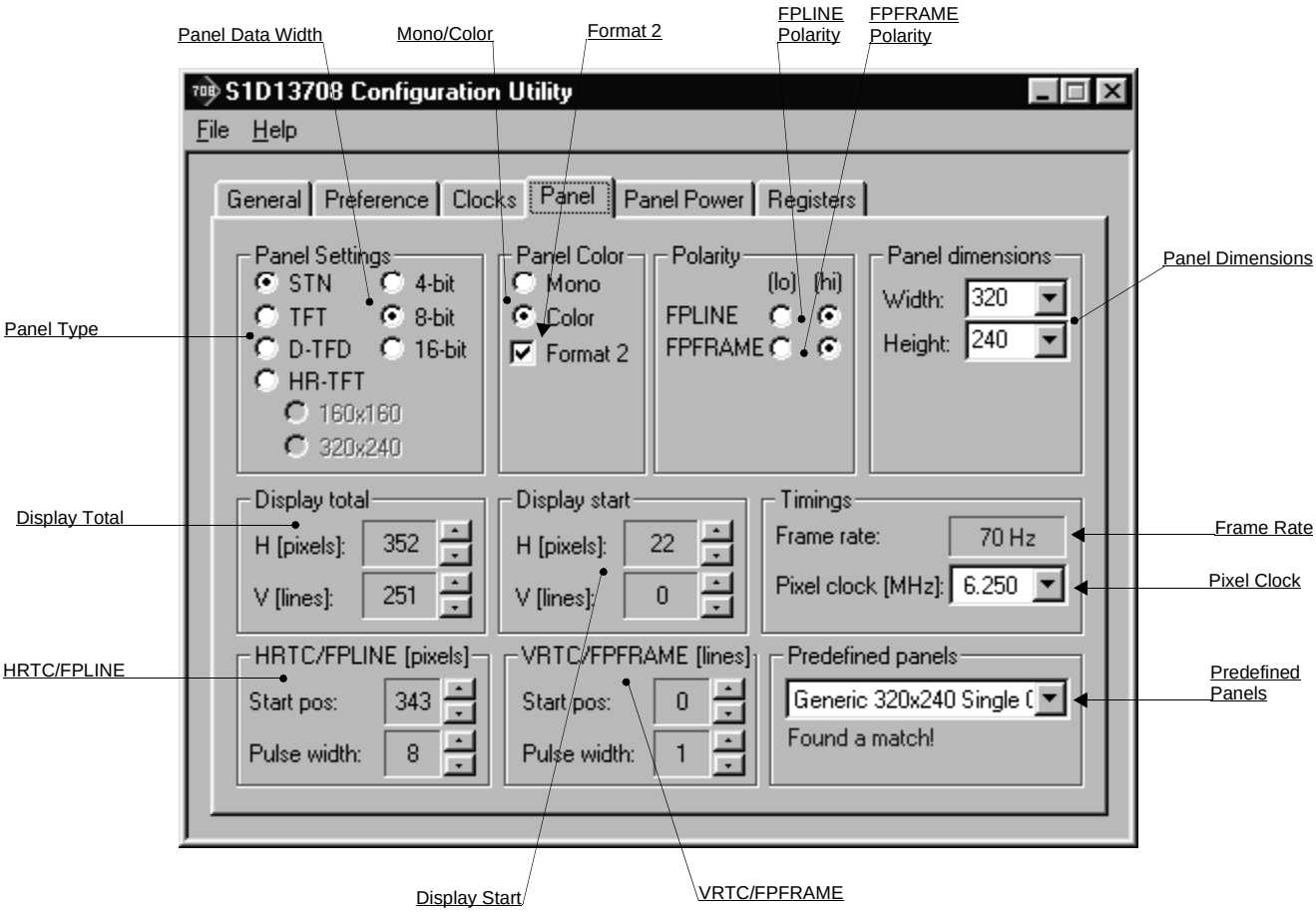
The signal CVOUT is forced high when this box is checked. When not checked, CVOUT will be low if CV is not enabled or will change state according to the configured values when CV is enabled.

Source

The CV Pulse uses the same source clock as the PWMCLK.

Divide	Specifies the divide ratio for the clock source. The divide ratio is applied to the CVOUT Pulse clock source to derive the CV Pulse clock frequency.
Timing	This field shows the actual CV Pulse frequency used by the configuration process.
Burst Length	The number of pulses generated in a single CV Pulse burst.

Panel Tab



The S1D13708 supports many panel types. This tab allows configuration of most panel related settings such as dimensions, type and timings.

Panel Type

Selects between passive (STN) and active (TFT/D-TFD/HR-TFT) panel types.

Some options may change or become unavailable when the STN/TFT/D-TFD/HR-TFT setting is changed. Reconfirm all settings on this tab after the Panel Type is changed.

Panel Data Width

Selects the panel data width. Panel data width is the number of bits of data transferred to the LCD panel on each clock cycle and shouldn't be confused with color depth which determines the number of displayed colors.

When a passive panel type (STN) is selected, the

available options are 4, 8, and 16 bit. When an active panel type (TFT/D-TFD/HR-TFT) is selected, the available options are 9, 12, and 18 bit.

Mono / Color

Selects between a monochrome or color panel.

Format 2

Selects color STN panel format 2. This option only applies to 8-bit color STN panels.

See the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx, for a description of format 1 / format 2 data formats. Most new panels use the format 2 data format.

Polarity

Allows selection of the polarity for the FPLINE and FPFRAME pulses.

Note
Selecting the wrong pulse polarity may damage the panel.

FPLINE Polarity

Selects the polarity of the FPLINE pulse.

Refer to the panel specification for the correct polarity of the FPLINE pulse.

FPFRAME Polarity

Selects the polarity of the FPFRAME pulse.

Refer to the panel specification for the correct polarity of the FPFRAME pulse.

Panel Dimensions

These fields specify the panel width and height. A number of common widths and heights are available in the selection boxes. If the width/height of your panel is not listed, enter the actual panel dimensions into the edit field.

For passive panels, manually entered pixel widths must be a minimum of 32 pixels and can be increased by multiples of 16. For TFT panels, manually entered pixel widths must be a minimum of 16 pixels and can be increased by multiples of 8. If a value is entered that does not match these requirements, a notification box appears and 13708CFG rounds up the value to the next allowable width.

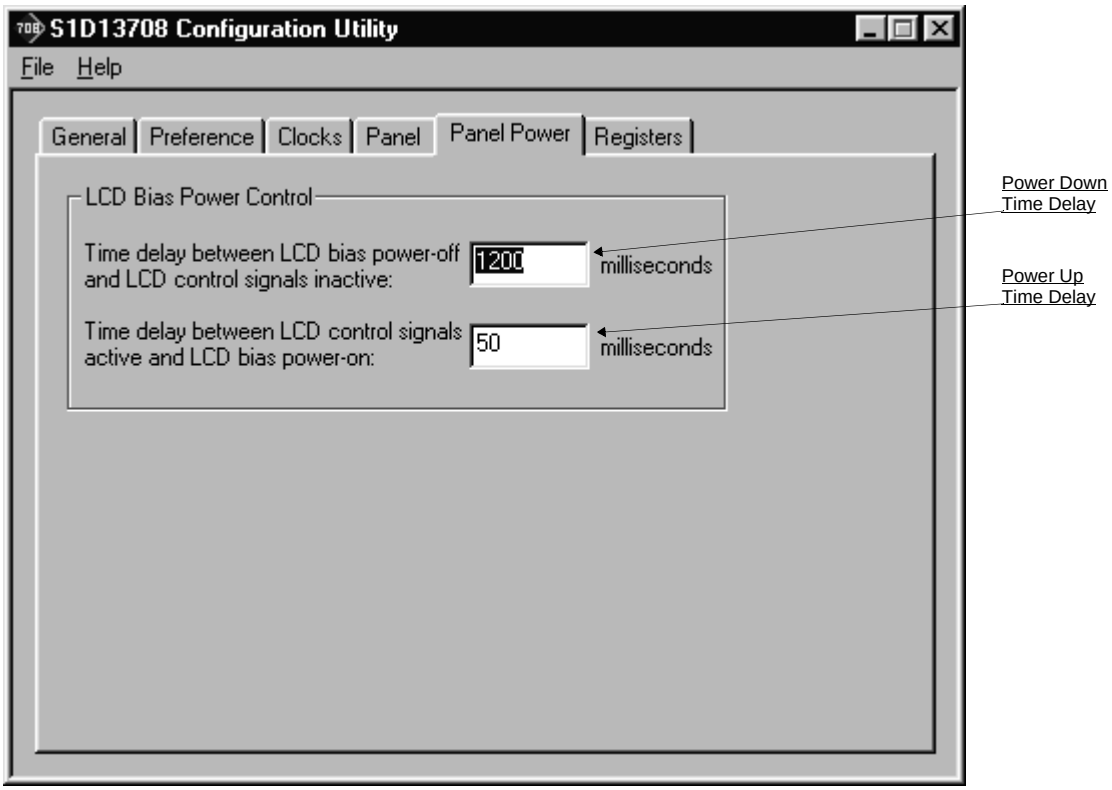
Display Total

It is recommended that the automatically generated Display Total values be used. However, manual adjustment may be used to improve the quality of the

	displayed image by fine tuning the horizontal and vertical display totals. The display total equals the display period plus the non-display period. Refer to *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx, for a complete description of the Display Total settings.
Display Start	It is recommended that the automatically generated Display Start values be used. However, manual adjustment may be used to improve the quality of the displayed image by fine tuning the horizontal and vertical display start positions. Refer to *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx, for a complete description of the Display Start settings.
Frame Rate	The Frame Rate (in Hz) is calculated and displayed based on the current settings as selected on the various tabs. If the resulting Frame Rate is not acceptable, adjust the settings to change the frame rate. Panel dimensions are fixed therefore frame rate can only be adjusted by changing either PCLK or display total values.
Pixel Clock	Select the desired Pixel Clock (in MHz) from the drop-down list. The range of frequencies displayed is dependent on the PCLK source and divide settings as selected on the Clocks tab.
HRTC/FPLINE (pixels)	These settings allow fine tuning of the TFT line pulse parameters. Refer to *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx for a complete description of the FPLINE pulse settings.
Start pos	Specifies the delay (in pixels) from the start of the horizontal non-display period to the leading edge of the FPLINE pulse.
Pulse Width	Specifies the width (in pixels) of the horizontal sync signal (FPLINE).

VRTC/FPFRAME (lines)	These settings allow fine tuning of the frame pulse parameters. Refer to *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx, for a complete description of the FPFRAME pulse settings.
Start pos	Specifies the delay (in lines) from the start of the vertical non-display period to the leading edge of the FPFRAME pulse.
Pulse width	Specifies the pulse width (in lines) of the vertical sync signal (FPFRAME).
Predefined Panels	13708CFG uses a file (**panels.def**) which contains predefined settings for a number of LCD panels. If the file **panels.def** is present in the same directory as **13708cfg.exe**, the predefined panels are available in the drop-down list. If a panel is selected from the list, 13708CFG loads the settings contained in the file.

Panel Power Tab



The S5U13708B00B evaluation board is designed to use the GPO0 signal to control the LCD bias power. The following settings configure panel power support.

Power Down Time Delay

This setting controls the minimum time delay between when the LCD panel is powered-off and when the S1D13708 control signals are turned off. This setting must be configured according to the specification for the panel being used.

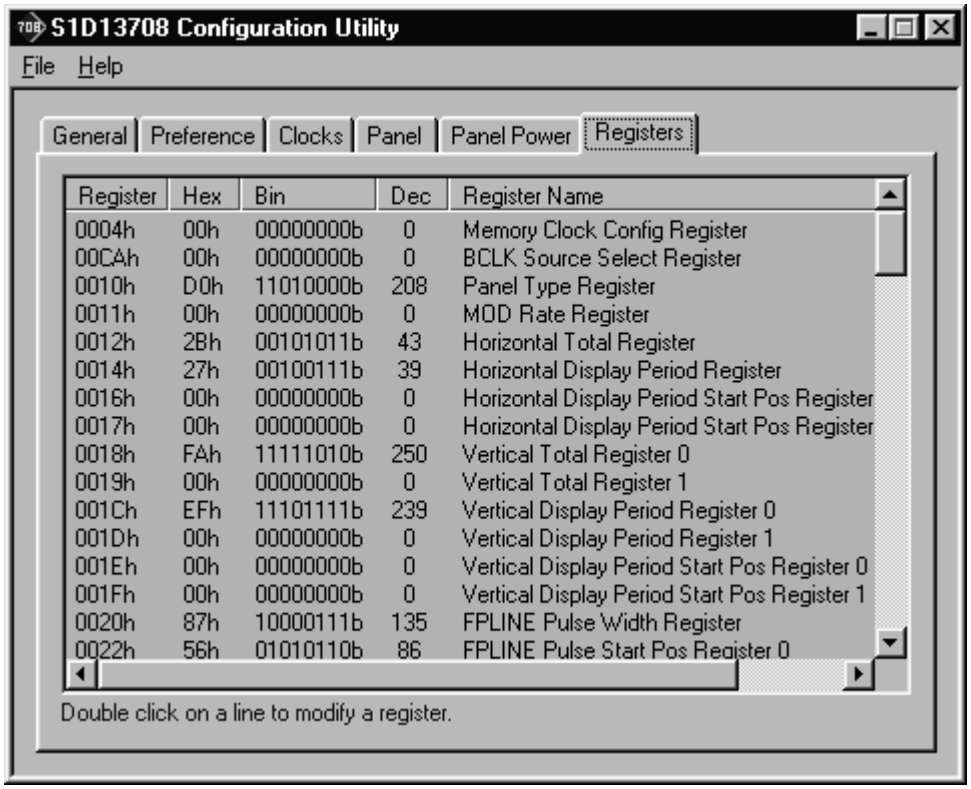
This value is used by Epson evaluation software designed for the S5U13708B00B evaluation board.

Power Up Time Delay

This setting controls the maximum time delay between when the S1D13708 control signals are turned on and the LCD panel is powered-on. This setting must be configured according to the specification for the panel being used.

This value is used by Epson evaluation software designed for the S5U13708B00B evaluation board.

Registers Tab



The Registers tab allows viewing and direct editing the S1D13708 register values.

Scroll up and down the list to view register values which are determined from the configuration settings of the previous tabs. Register settings may be changed by double-clicking on the register in the listing. **Manual changes to the registers are not checked for errors, so caution is warranted when directly editing these values.** It is strongly recommended that the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx be referred to before making any manual register settings.

Note

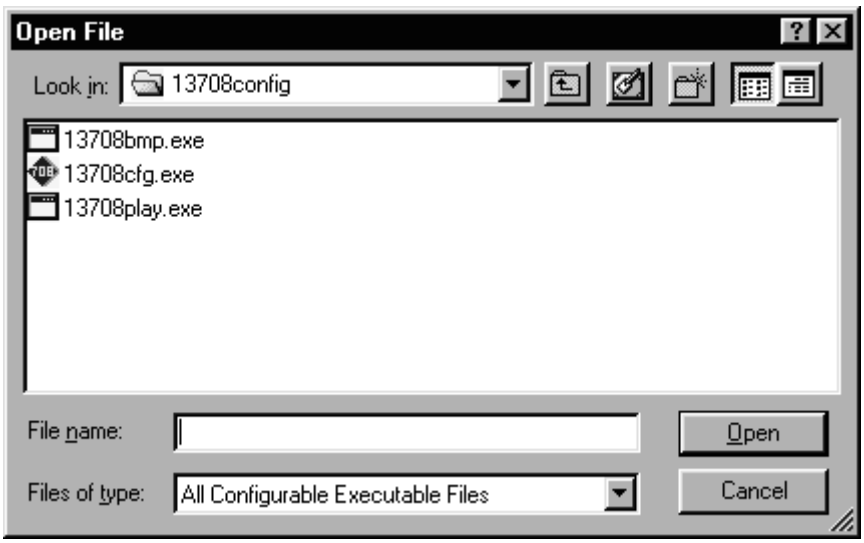
Manually entered values may be changed by 13708CFG if further configuration changes are made on other tabs. In this case, the user is notified.

13708CFG Menus

The following sections describe each of the options in the File and Help menus.

Open...

From the Menu Bar, select “File”, then “Open...” to display the Open File Dialog Box.



The Open option allows 13708CFG to read the configuration information from programs based on the HAL library. When 13708CFG opens a file it scans the file for an identification string, and if found, reads the configuration information. This feature may be used to quickly arrive at a starting point for register configuration. The only requirement is that the file being opened must contain a valid S1D13708 HAL library information block.

13708CFG supports a variety of executable file formats. Select the file type(s) 13708CFG should display in the Files of Type drop-down list and then select the filename from the list and click on the Open button.

Note

13708CFG is designed to work with utilities programmed using a given version of the HAL. If the configuration structure contained in the executable file differs from the version 13708CFG expects the Open will fail and an error message is displayed. This may happen if the version of 13708CFG is substantially older, or newer, than the file being opened.

Save

From the Menu Bar, select “File”, then “Save” to initiate the save action. The Save menu option allows a fast save of the configuration information to a file that was opened with the Open menu option.

Note
This option is only available once a file has been opened.

Save As...

From the Menu Bar, select “File”, then “Save As...” to display the Save As Dialog Box.

“Save as” is very similar to Save except a dialog box is displayed allowing the user to name the file before saving.

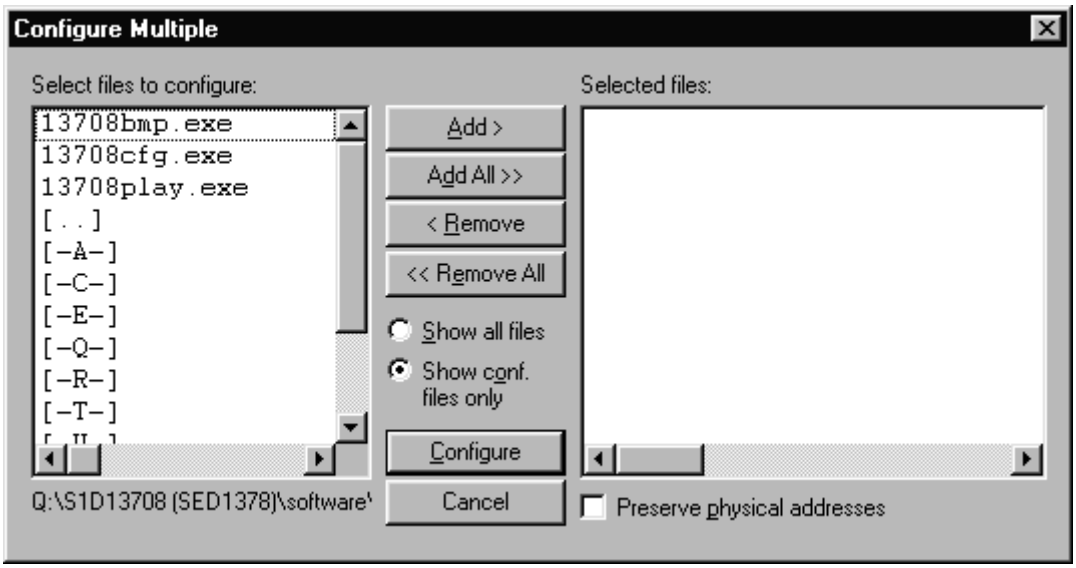
Using this technique a tester can configure a number of files differing only in configuration information and name (e.g. BMP60Hz.EXE, BMP72Hz.EXE, BMP75Hz.EXE where only the frame rate changes in each of these files).

Note
When “Save As” is selected then an exact duplicate of the file as opened by the “Open” option is created containing the new configuration information.

Configure Multiple

After determining the desired configuration, “Configure Multiple” allows the information to be saved into one or more executable files built with the HAL library.

From the Menu Bar, select “File”, then “Configure Multiple” to display the Configure Multiple Dialog Box. This dialog box is also displayed when a file(s) is dragged onto the 13708CFG window.



The left pane lists files available for configuration; the right pane lists files that have been selected for configuration. Files can be selected by clicking the “Add” or “Add All” buttons, double clicking any file in the left pane, or by dragging the file(s) from Windows Explorer.

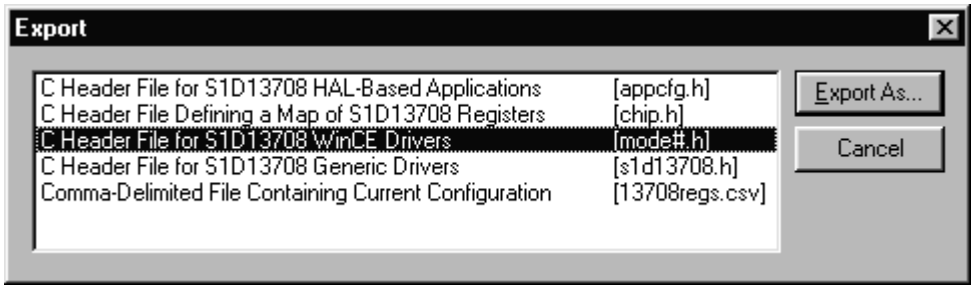
Selecting “Show all files” displays all files in the selected directory, whereas selecting “Show conf. files only” will display only files that can be configured using 13708CFG (i.e. .exe, .elf, .s9, .s19, .idp, .lvb).

Checking “Preserve Physical Addresses” instructs 13708CFG to use the register and display buffer address values the files were previously configured with. Addresses specified in the General Tab are discarded. This is useful when configuring several programs for various hardware platforms at the same time. For example, if configuring PCI and S19 files at the same time for a new panel type, the physical addresses for each are retained. This feature is primarily intended for the test lab where multiple hardware configurations exist and are being tested.

Export

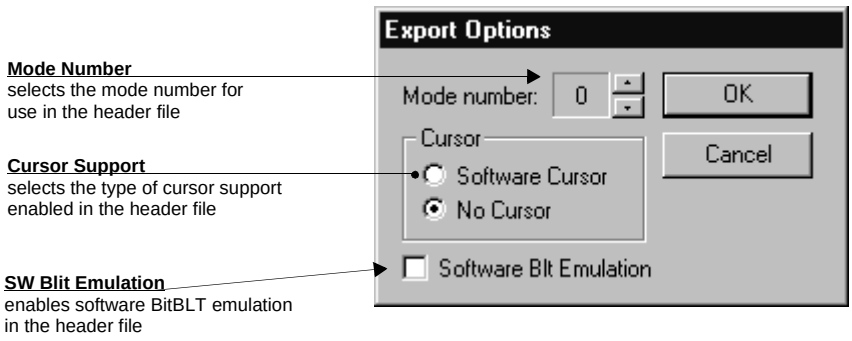
After determining the desired configuration, “Export” permits the user to save the register information as a variety of ASCII text file formats. The following is a list and description of the currently supported output formats:

- a C header file for use in writing HAL library based applications.
- a C header file which lists each register and the value it should be set to.
- a C header file for use in developing Window CE display drivers.
- a C header file for use in developing display drivers for other operating systems such as Linux, QNX, and VxWorks UGL or WindML.
- a comma delimited text file containing an offset, a value, and a description for each S1D13708 register.



After selecting the file format, click the “Export As..." button to display the file dialog box which allows the user to enter a filename before saving. Clicking the “Preview” button uses Notepad or the web browser to display a copy of the file to be saved.

When the **C Header File for S1D13708 WinCE Drivers** option is selected as the export type, additional options are available and can be selected by clicking on the Options button. The options dialog appears as:



Enable Tooltips

Tooltips provide useful information about many of the items on the configuration tabs. Placing the mouse pointer over nearly any item on any tab generates a popup window containing helpful advice and hints.

To enable/disable tooltips check/uncheck the “Tooltips” option form the “Help” menu.

Note
Tooltips are enabled by default.

ERD on the Web

This “Help” menu item is actually a hotlink to the Epson Research and Development website. Selecting “Help” then “ERD on the Web” starts the default web browser and points it to the ERD product web site.

The latest software, drivers, and documentation for the S1D13708 is available at this website.

About 13708CFG

Selecting the “About 13708CFG” option from the “Help” menu displays the About dialog box for 13708CFG. The about dialog box contains version information and the copyright notice for 13708CFG.

Comments

- On any tab particular options may be grayed out if selecting them would violate the operational specification of the S1D13708 (i.e. Selecting TFT or STN on the Panel tab enables/disables options specific to the panel type).
- The file **panels.def** is a text file containing operational specifications for several supported, and tested, panels. This file can be edited with any text editor.
- 13708CFG allows manually altering register values. The manual changes may violate memory and LCD timings as specified in the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx. If this is done, unpredictable results may occur. Epson Research and Development, Inc. does not assume liability for any damage done to the display device as a result of configuration errors.

EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

13708PLAY Diagnostic Utility

Document Number: X39A-B-002-01

THIS PAGE LEFT BLANK

13708PLAY

13708PLAY is a diagnostic utility which allows a user to read/write all the registers and display buffer of the S1D13708. Commands are received from the standard input device, and messages are sent to the standard output device. On Intel platforms the console provides standard input/output. For most embedded systems the serial device provides these functions.

Commands can be entered interactively by a user, or be executed from a script file. Scripting is a powerful feature which allows command sequences to be used repeatedly without re-entry.

S1D13708 Supported Evaluation Platforms

13708PLAY is available as an executable for PCs running Windows® 9x/ME/NT/2000 and as C source code which can be modified and recompiled to allow 13708PLAY to run on other evaluation platforms.

Installation

PC platform

Copy the file **13708play.exe** to a directory in the path (e.g. PATH=C:\S1D13708).

Embedded platform

Download the program **13708play** to the system.

Usage

PC platform

At the prompt, type:

```
13708play [/?]
```

Where:

/? displays copyright and program version information.

Embedded platform

Execute **`13708play`** and at the prompt, type the command line argument **/?**.

Where:

/? displays copyright and program version information.

Commands

The following commands are intended to be used from within the 13708PLAY program. However, simple commands can also be executed from the command line (e.g. **13708PLAY F 0 13FFF 0**).

Note

If the host platform is big endian, reading/writing words and dwords to/from the registers and display buffer may result in unexpected values. In this case it will be necessary for the user to manually swap the bytes in order to perform the IO correctly. For further information on little/big endian and the S1D13708 byte/word swapping capabilities, see the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

CLKI freq
Sets the frequency of CLKI.

Where:

freq	The desired frequency for CLKI (in MHz).

CLKI2 freq
Sets the frequency of CLKI2.

Where:

freq	The desired frequency for CLKI2 (in MHz).

D[S][8|16|32] [startaddr [endaddr|len]]
Displays a memory dump from the specified display buffer address range.

Where:

S	Sets the base address to the beginning of system memory. Without the “S” option, the base address is the beginning of display buffer memory.
8|16|32	The unit size: 8-bit (bytes), 16-bit (words), 32-bit (dwords). If a unit size is not specified, this command uses the unit size from the last Dump command performed. If no previous Dump command has been issued, the unit size defaults to 8-bit.
startaddr	The starting address to read data from. Specifying a period (.) uses the same starting address as the last Dump command performed. Specifying a *startaddr* of two periods (..) will back the start address by the size of *len*.
endaddr|len	Determines how many units to continue dumping the contents of the display buffer. A number without a prefix represents a physical ending address. If an “L” prefix is used, the number that follows represents *len*, which is the number of bytes/words/dwords to be dumped. *Len* is based on the unit size. For example, 'L8' when the unit size is 16-bit would cause the Dump command to dump 8 words from the starting address.

F[S][8|16|32] startaddr endaddr|len data1 [data2 data3 ...]
Fills a specified address range in the display buffer.

Where:

S	Sets the base address to the beginning of system memory. Without the "S" option, the base address is the beginning of display buffer memory.
8|16|32	The unit size: 8-bit (bytes), 16-bit (words), 32-bit (dwords). If a unit size is not specified, this command uses the unit size from the last Fill command performed. If no previous Fill command has been issued, the unit size defaults to 8-bit.
startaddr	The starting address to begin filling at. Specifying a period (.) uses the same starting address as the last Fill command performed.
endaddr|len	Determines how many units to fill the display buffer with. A number without a prefix represents a physical ending address. If a "L" prefix is used, the number that follows represents *len*, or the number of bytes/words/dwords to be filled. *Len* is based on the unit size. For example, 'L8' when the unit size is 16-bit would cause the Fill command to fill 8 words from the starting address.
data	The value(s) used to fill the display buffer. If multiple values are given, the pattern repeats through memory. Values can be combinations of 'text' or numbers. Numbers are assumed to be hexadecimal values unless otherwise specified with the correct suffix (binary = i, octal = o, decimal = t, hexadecimal = h). For example, 101i = 101 binary.

H [lines]
Sets the number of lines of data that are displayed at a time. The display is halted after the specified number of lines. Setting the number of lines to 0 disables the halt function and allows the data to continue displaying until all data has been shown.

This command is useful when large blocks of the display buffer or the contents of the LUT are being viewed.

Where:

lines	Number of lines that are shown before halting the displayed data (decimal value).

I
Initializes the S1D13708 registers with the default register settings as configured by the utility 13708CFG. For further information on 13708CFG, see the *13708CFG User Manual*, document number X39A-B-001-xx.

Note
If the "I" command is used before 13708PLAY is configured, an error message is displayed and no action is taken.

L index [red green blue]
Reads/writes the red, green, and blue Look-Up Table (LUT) components. If the red, green, and blue components are not specified, the LUT for the given index is read and the RGB values are displayed.

Where:

index	Index into the LUT (hex).
red	Red component of the LUT (hex).
green	Green component of the LUT (hex).
blue	Blue component of the LUT (hex).

Note
Only bits 7-2 of each color are used in the LUT. The least significant two bits of the colors are discarded. For example, the command **L 0 1 2 3** will set each RGB component of LUT index 0 to 0, as the values 1, 2, an 3 use only the least significant bits.

LA
Reads and displays all LUT values.

M [bpp]
Sets the color depth (bpp). If no color depth is provided, information about the current settings are displayed.

Where:

bpp	Color depth to be set (1/2/4/8/16 bpp).

Note
This command reads and attempts to interpret the S1D13708 control registers. If the S1D13708 was not correctly initialized the displayed values may be incorrect.

Q[UIT], EXIT
Quits the program.

R[S][8|16|32] [addr1 addr2 addr3 ...]
Reads the display buffer at the address locations given.

Where:

S	Sets the base address to the beginning of system memory. Without the 'S' option, the base address is the beginning of display buffer memory.
8|16|32	The unit size: 8-bit (bytes), 16-bit (words), 32-bit (dwords). If a unit size is not specified, this command uses the unit size from the last Read command performed. If no previous Read command has been issued, the unit size defaults to 8-bit.
addr	The address to read data from. Multiple addresses can be given.

RI [8|16] [cycles]
For testing the indirect interface ONLY.
RI will issue one or more data read cycles from the indirect interface cmd/data cycle.

Where:

8|16	The unit size: 8-bit (bytes), 16-bit (words). If a unit size is not specified, this command uses the unit size from the last Read Indirect command performed. If no previous Read Indirect command has been issued, the unit size defaults to 8-bit.
cycles	The number of data read cycles. The default is one cycle.

RUN scriptfile
This command opens the file scriptfile and executes each line as if it were typed from the command prompt. For more information on scriptfiles, see **Section , *"Script Files"* on page 12**.

Where:

scriptfile	The file containing 13708PLAY commands.

S[S][8|16|32] startaddr endaddr|len data1 [data2 data3 data4 ...]
Search the display buffer for the given data.

Where:

S	Sets the base address to the beginning of system memory. Without the "S" option, the base address is the beginning of display buffer memory.
8|16|32	The unit size: 8-bit (bytes), 16-bit (words), 32-bit (dwords). If a unit size is not specified, this command uses the unit size from the last Search command performed. If no previous Search command has been issued, the unit size defaults to 8-bit.
startaddr	The starting address to begin the search from. Specifying a period (.) uses the same starting address as the last Search command performed.
endaddr|len	Determines how many units of the display buffer will be searched through. A number without a prefix represents a physical ending address. If a "L" prefix is used, the number that follows represents *len*, or the number of bytes/words/dwords to be searched through. *Len* is based on the unit size. For example, 'L8' when the unit size is 16-bit would cause the Search command to search 8 words from the starting address.
data	The value(s) to search the display buffer for. Values can be combinations of 'text' or numbers. Numbers are assumed to be hexadecimal values unless otherwise specified with the correct suffix (binary = i, octal = o, decimal = t, hexadecimal = h). For example, 101i = 101 binary.

SHOW
Shows a test pattern on the display. The test pattern is based on current register settings and may not display correctly if the registers are not configured properly.

Use this command to display an image during testing. After adjusting a register value, use the show command to view the effect on the display.

W[S][8|16|32] [startaddr [data1 data2 data3 data4 ...]]
Writes the given data sequence to the display buffer starting at startaddr location.

Where:

S	Sets the base address the beginning of system memory. Without the "S" option, the base address is the beginning of display buffer memory.
8|16|32	The unit size: 8-bit (bytes), 16-bit (words), 32-bit (dwords). If a unit size is not specified, this command uses the unit size from the last Write command performed. If no previous Write command has been issued, the unit size defaults to 8-bit.
startaddr	The starting address to write data to. Specifying a period (.) uses the same starting address as the last Write command performed.
data	Values to write to the display buffer. If no data is given, then this function enters MODIFY mode. This mode prompts the user with the address and it's current data. While in this mode, the user can type any of the following. - new values (in hex) - ENTER or SPACE - moves to the next memory location (if data is specified the previous memory location is updated, if no data is specified no change is made) - "-" - moves to the previous memory location - "Q" or "." - exits MODIFY mode

WI [8|16] data1 [data2 data3 ...]
For testing the indirect interface ONLY.
WI will write one or more data items to the data portion of the cmd/data cycle.

Where:

8|16	The unit size: 8-bit (bytes), 16-bit (words). If a unit size is not specified, this command uses the unit size from the last Read Indirect command performed. If no previous Read Indirect command has been issued, the unit size defaults to 8-bit.
data	Values to write to the display buffer.

XA
Reads and displays the contents of all the S1D13708 registers.

X[8|16|32] [index [data]]
Reads/writes data to the register at index. If no data is specified, the register is read and the contents are displayed.

Where:

8|16|32	The unit size: 8-bit (bytes), 16-bit (words), 32-bit (dwords). If a unit size is not specified, this command uses the unit size from the last X command performed. If no previous X command has been issued, the unit size defaults to 8-bit.
index	Index into the registers (hex).
data	The value to be written to the register. Numbers are assumed to be hexadecimal values unless otherwise specified with the correct suffix (binary = i, octal = o, decimal = t, hexadecimal = h). For example, 101i = 101 binary.

XI index
This command is for testing the indirect interface ONLY.
XI will write index to the command portion of the indirect interface cmd/data cycle. The data portion will NOT be accessed.

Where:

index	The command portion of the indirect interface cmd/data cycle. This is effectively the register index (hex).

?
Displays the help screen. The help screen contains a summary of all available commands.

13708PLAY Example

1. Configure **13708PLAY** using the utility **13708CFG**. For further information on 13708CFG, see the *13708CFG User Manual*, document number X39A-B-001-xx.
2. Type **13708PLAY** to start the program.
3. Type **?** for help.
4. Type **i** to initialize the registers.
5. Type **xa** to display the contents of the registers.
6. Type **x 80** to read register 80h.
7. Type **x 80 10** to write 10h to register 80h.
8. Type **f 0 ffff aa** to fill the first 10000h bytes of the display buffer with AAh.
9. Type **d 0 ff** to read the first 100h bytes of the display buffer.
10. Type **show** to display a test pattern.
11. Type **m** to display current mode information.
12. Type **m 2** to set the color depth to 2 bpp.
13. Type **show** to display a test pattern.
14. Type **q** to exit the program.

Script Files

13708PLAY can be controlled by a script file. This is useful when:

- there is no display to monitor command keystroke accuracy.
- various registers must be quickly changed to view results.

A script file is an ASCII text file with one 13708PLAY command per line. Script files can be executed from within 13708PLAY using the Run command (e.g. = run dumpregs.scr). Alternately, the script file can be executed from the OS command prompt. On a PC platform, a typical script command line might be:

13708PLAY run dumpregs.scr > results

This causes the file **dumpregs.scr** to be interpreted as commands by 13708PLAY and the results to be redirected to the file **results**.

Example 1: Create a script file that reads all registers.

```
; This file initializes the S1D13708 and reads the registers.
; Note: after a semicolon (;), all characters on a line are ignored.

; Initialize the S1D13708
i

; Read all registers
xa
```

EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

13708BMP Demonstration Program

Document Number: X39A-B-003-01

THIS PAGE LEFT BLANK

13708BMP

13708BMP is a demonstration utility used to show the S1D13708 display capabilities by rendering bitmap images on the display device. The program displays a bitmap stored in Windows BMP file format and then exits. 13708BMP supports SviwelView™ (90°, 180°, and 270° hardware rotation of the display image).

13708BMP is designed to operate on a personal computer (PC) within a 32-bit environment only (Windows® 9x/NT/ME/2000). Embedded platforms are not supported due to the possible lack of a structured file system.

The 13708BMP demonstration utility must be configured to work with your hardware configuration. The program 13708CFG can be used to configure 13708BMP. For further information on 13708CFG, refer to the *13708CFG Users Manual*, document number X39A-B-001-xx.

S1D13708 Supported Evaluation Platforms

13708BMP supports the following S1D13708 evaluation platforms:

- PC with an Intel 80x86 processor running Windows 9x/NT/ME/2000.

Note
The 13708BMP source code may be modified by the OEM to support other evaluation platforms.

Installation

Copy the file **13708bmp.exe** to a directory in the path (e.g. PATH=C:\S1D13708).

Usage

At a command prompt, type:

13708bmp bmpfile [/?]

Where:

bmpfile	Specifies filename of the windows format bmp image to be displayed.
/?	Displays the help message.

Note
13708BMP displays the bmpfile image and returns to the prompt.

13708BMP Examples

To display a .bmp image in the main window on an LCD, type the following:
13708bmp bmpfile.bmp

Note
To display a bmpfile using SwivelView, configure 13708bmp.exe for the selected SwivelView mode using the configuration program 13708CFG. For further information on 13708CFG, see the *13708CFG User Manual*, document number X39A-B-001-xx.

Comments

- 13708BMP displays only Windows BMP format images.
- A 24-bit true color bitmap is displayed at a color depth of 16 bit-per-pixel.
- Only the green component of the image is seen on a monochrome panel.
- 13708BMP does not perform any image translations. The image to be displayed must be the desired dimensions and color depth.

S1D13708 Embedded Memory LCD Controller

Wind River WindML v2.0 Display Drivers

Document Number: X39A-E-002-01

THIS PAGE LEFT BLANK

Wind River WindML v2.0 DISPLAY DRIVERS

The Wind River WindML v2.0 display drivers for the S1D13708 Embedded Memory LCD Controller are intended as "reference" source code for OEMs developing for Wind River's WindML v2.0. The driver package provides support for both 8 and 16 bit-per-pixel color depths. The source code is written for portability and contains functionality for most features of the S1D13708. Source code modification is required to produces a smaller, more efficient driver for mass production.

The WindML display drivers are designed around a common configuration include file called **mode0.h** which is generated by the configuration utility 13708CFG. This design allows for easy customization of display type, clocks, decode addresses, rotation, etc. by OEMs. For further information on 13708CFG, see the *13708CFG Configuration Program User Manual*, document number X39A-B-001-xx.

Note

The WindML display drivers are provided as "reference" source code only. They are intended to provide a basis for OEMs to develop their own drivers for WindML v2.0. These drivers are not backwards compatible with UGL v1.2.

This document and the source code for the WindML display drivers is updated as appropriate. Please check the Epson Research and Development website at www.erd.epson.com for the latest revisions before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

Building a WindML v2.0 Display Driver

The following instructions produce a bootable disk that automatically starts the UGL demo program. These instructions assume that Wind River's Tornado platform is already installed.

Note
For the example steps where the drive letter is given as "x:". Substitute "x" with the drive letter that your development environment is on.

1. Create a working directory and unzip the WindML display driver into it.

   From a command prompt or GUI interface create a new directory (e.g. x:\13708).

   Unzip the file **13708windml.zip** to the newly created working directory. The files will be unzipped to the directories "x:\13708\8bpp" and "x:\13708\16bpp".

2. Configure for the target execution model.

   This example build creates a VxWorks image that fits onto and boots from a single floppy diskette. In order for the VxWorks image to fit on the disk certain modifications are required.

   Replace the file "x:\Tornado\target\config\pcPentium\config.h" with the file "x:\13708\8bpp\File\config.h" (or "x:\13708\16bpp\File\config.h"). The new **config.h** file removes networking components and configures the build image for booting from a floppy disk.

Note
Rather than simply replacing the original **config.h** file, rename it so the file can be kept for reference purposes.

3. Build a boot ROM image.

   From the Tornado tool bar, select Build -> Build Boot ROM. Select "pcPentium" as the BSP and "bootrom_uncmp" as the image.

4. Create a bootable disk (in drive A:).

   From a command prompt change to the directory "x:\Tornado\host\x86-win32\bin" and run the batch file **torvars.bat**. Next, change to the directory "x:\Tornado\target\config\pcPentium" and type:

   mkboot a: bootrom_uncmp

5. If necessary, generate a new **mode0.h** configuration file.

   The file **mode0.h** contains the register values required to set the screen resolution, color depth (bpp), display type, rotation, etc. The **mode0.h** file included with the drivers, may not contain applicable values and must be regenerated. The configuration program 13708CFG can be used to build a new **mode0.h** file. If building for 8 bpp, place the new **mode0.h** file in the directory "x:\13708\8bpp\File". If building for 16 bpp, place the new **mode0.h** file in "x:\13708\16bpp\File".

Note

Mode0.h should be created using the configuration utility 13708CFG. For more information on 13708CFG, see the *13708CFG Configuration Program User Manual*, document number X39A-B-001-xx available at www.erd.epson.com.

6. Build the WindML v2.0 library.

   From a command prompt change to the directory "x:\Tornado\host\x86-win32\bin" and run the batch file **torvars.bat**. Next, change to the directory "x:\Tornado\target\src\ugl" and type the command:

   make CPU=PENTIUM ugl

7. Open the S1D13708 workspace.

   From the Tornado tool bar, select File->Open Workspace...->Existing->Browse... and select the file "x:\13708\8bpp\13708.wsp" (or "x:\13708\16bpp\13708.wsp").

8. Add support for single line comments.

   The WindML v2.0 display driver source code uses single line comment notation, "//", rather than the ANSI conventional comments, "/*...*/".

   To add support for single line comments follow these steps:

   a. In the Tornado "Workspace Views" window, click on the "Builds" tab.

   b. Expand the "8bpp Builds" (or "16bpp Builds") view by clicking on the "+" next to it. The expanded view will contain the item "default". Right-click on "default" and select "Properties...". A "Properties:" window will appear.

   c. Select the "C/C++ compiler" tab to display the command switches used in the build. Remove the "-ansi" switch from the line that contains "-g -mpentium -ansi -nostdinc -DRW_MULTI_THREAD".
   (Refer to GNU ToolKit user's guide for details)

9. Compile the VxWorks image.

   Select the "Builds" tab in the Tornado "Workspace Views" window.

   Right-click on "8bpp files" (or "16bpp files") and select "Dependencies...". Click on "OK" to regenerate project file dependencies for "All Project files".

   Right-click on "8bpp files" (or "16bpp files") and select "ReBuild All(vxWorks)" to build VxWorks.

10. Copy the VxWorks file to the diskette.

    From a command prompt or through the Windows interface, copy the file "x:\13708\8bpp\default\vxWorks" (or "x:\13708\16bpp\default\vxWorks") to the bootable disk created in step 4.

11. Start the VxWorks demo.

    Boot the target PC with the VxWorks bootable diskette to run the UGL demo program automatically.

THIS PAGE LEFT BLANK

S1D13708 Embedded Memory LCD Controller

Linux Console Driver

Document Number: X39A-E-004-01

THIS PAGE LEFT BLANK

Linux Console Driver

The Linux console driver for the S1D13708 Embedded Memory LCD Controller is intended as "reference" source code for OEMs developing for Linux, and supports 4, 8, and 16 bit-per-pixel color depths. The source code is written for portability and contains functionality for most features of the S1D13708. Source code modification is required to provide a smaller driver for mass production.

A Graphical User Interface (GUI) such as Gnome can obtain the frame buffer address from this driver allowing the Linux GUI the ability to update the display.

The console driver is designed around a common configuration include file called **s1d13708.h,** which is generated by the configuration utility 13708CFG. This design allows for easy customization of display type, clocks, decode addresses, rotation, etc. by OEMs. For further information on 13708CFG, see the *13708CFG Configuration Program User Manual*, document number X39A-B-001-xx.

Note

The Linux console driver is provided as "reference" source code only. The driver is intended to provide a basis for OEMs to develop their own drivers for Linux.

This document and the source code for the Linux console drivers are updated as appropriate. Please check the Epson Research and Development website at www.erd.epson.com for the latest revisions or before beginning any development.

We appreciate your comments on our documentation. Please contact us via e-mail at documentation@erd.epson.com.

Building the Console Driver for Linux Kernel 2.2.x

Follow the steps below to construct a copy of the Linux operating system using the S1D13708 as the console display device. These instructions assume that the GNU development environment is installed and the user is familiar with GNU and the Linux operating system.

1. Acquire the Linux kernel source code.

   You can obtain the Linux kernel source code from your Linux supplier or download the source from: ftp://ftp.kernel.org.

   The S1D13708 reference driver requires Linux kernel 2.2.x. The example S1D13708 reference driver available on www.erd.epson.com was built using Red Hat Linux 6.1, kernel version 2.2.17.

   For information on building the kernel refer to the readme file at:
   ftp://ftp.linuxberg.com/pub/linux/kernel/README

Note
Before continuing with modifications for the S1D13708, you should ensure that you can build and start the Linux operating system.

2. Unzip the console driver files.

   Using a zip file utility, unzip the S1D13708 archive to a temporary directory. (e.g. /tmp)

   When completed the following files should be located in the temporary directory.
   s1d13xxxfb.c
   s1d13708.h
   Config.in
   fbmem.c
   fbcon-cfb4.c, and
   Makefile

3. Copy the console driver files to the build directory.

   Copy the following files to the directory /usr/src/linux/drivers/video.
   /tmp/s1d13xxxfb.c and
   /tmp/s1d13708.h

   Copy the remaining source files into the directory /usr/src/linux/drivers/video replacing the files of the same name.
   /tmp/Config.in
   /tmp/fbmem.c
   /tmp/fbcon-cfb4.c, and
   /tmp/Makefile

   If your kernel version is not 2.2.17, or you want to retain greater control of the build process, use a text editor to cut and paste the sections dealing with the Epson driver in the corresponding files of the same names.

4. Modify s1d13708.h

   The file s1d13708.h contains the register values required to set the screen resolution, color depth (bpp), display type, active display (LCD), display rotation, etc.

   Before building the console driver, refer to the descriptions in the file s1d13708.h for the default settings of the console driver. If the default does not match the configuration you are building for then s1d13708.h must be regenerated with the correct information.

   Use the program 13708CFG to generate the required header file. For information on how to use 13708CFG, refer to the *13708CFG Configuration Program User Manual*, document number X39A-B-001-xx, available at www.erd.epson.com

   After selecting the desired configuration, choose "File->Export" and select the "C Header File for S1D13708 Generic Drivers" option. Save the new configuration as s1d13708.h in the /usr/src/linux/drivers/video, replacing the original configuration file.

5. Configure the video options.

   From the command prompt in the directory /usr/src/linux run the command:
   make menuconfig

   This command will start a text based interface which allows the selection of build time parameters. From the text interface under "Console drivers" options, select:

   "Support for frame buffer devices"
   "Epson LCD/CRT controllers support"
   "S1D13708 support"
   "Advanced low level driver options"
   "xBpp packed pixels support" *

   * where x is the color depth being compile for.

   If you are using the Epson PCI evaluation board then you must also select:

   "Epson PCI Bridge adapter support"

   Once you have configured the kernel options, save and exit the configuration utility.

6. Compile and install the kernel.

   Build the kernel with the following sequence of commands.
   make dep
   make clean
   make bzImage
   /sbin/lilo (if running lilo)

7. Boot to the Linux operating system.

   If you are using lilo (Linux Loader), modify the lilo configuration file as discussed in the kernel build README file. If there were no errors during the build, from the command prompt run:

   ```
   lilo
   ```

   and reboot your system.

Note

In order to use the S1D13708 console driver with X server, you need to configure the X server to use the FBDEV device. The information on the necessary files and instructions for this process is available on the Internet at www.xfree86.org

Building the Console Driver for Linux Kernel 2.4.x

Follow the steps below to construct a copy of the Linux operating system using the S1D13708 as the console display device. These instructions assume that the GNU development environment is installed and the user is familiar with GNU and the Linux operating system.

1. Acquire the Linux kernel source code.

   You can obtain the Linux kernel source code from your Linux supplier or download the source from: ftp://ftp.kernel.org.

   The S1D13708 reference driver requires Linux kernel 2.4.x or greater. The example S1D13708 reference driver available on www.erd.epson.com was built using Red Hat Linux 6.1, kernel version 2.4.5.

   For information on building the kernel refer to the readme file at:
   ftp://ftp.linuxberg.com/pub/linux/kernel/README

Note
Before continuing with modifications for the S1D13708, you should ensure that you can build and start the Linux operating system.

2. Unzip the console driver files.

   Using a zip file utility, unzip the S1D13708 archive to a temporary directory. (e.g. /tmp)

   When completed the files:
   Config.in
   fbmem.c
   fbcon-cfb4.c
   Makefile
   should be located in the temporary directory (/tmp), and the files:
   Makefile
   s1d13xxxfb.c
   s1d13708.h
   should be located in a sub-directory called epson within the temporary directory (/tmp/epson).

3. Copy the console driver files to the build directory. Make the directory /usr/src/linux/drivers/video/epson.

   Copy the files
   /tmp/epson/s1d13xxxfb.c
   /tmp/epson/s1d13708.h
   /tmp/epson/Makefile
   to the directory /usr/src/linux/drivers/video/epson.

Copy the remaining source files
/tmp/Config.in
/tmp/fbmem.c
/tmp/fbcon-cfb4.c
/tmp/Makefile
into the directory /usr/src/linux/drivers/video replacing the files of the same name.

If your kernel version is not 2.4.5 or you want to retain greater control of the build process then use a text editor and cut and paste the sections dealing with the Epson driver in the corresponding files of the same names.

4. Modify s1d13708.h

   The file s1d13708.h contains the register values required to set the screen resolution, color depth (bpp), display type, active display (LCD/CRT), display rotation, etc.

   Before building the console driver, refer to the descriptions in the file s1d13708.h for the default settings of the console driver. If the default does not match the configuration you are building for then s1d13708.h will have to be regenerated with the correct information.

   Use the program 13708CFG to generate the required header file. For information on how to use 13708CFG, refer to the *13708CFG Configuration Program User Manual*, document number X39A-B-001-xx, available at www.erd.epson.com

   After selecting the desired configuration, choose “File->Export” and select the “C Header File for S1D13708 Generic Drivers” option. Save the new configuration as s1d13708.h in the /usr/src/linux/drivers/video, replacing the original configuration file.

5. Configure the video options.

   From the command prompt in the directory /usr/src/linux run the command:
   make menuconfig

   This command will start a text based interface which allows the selection of build time parameters. From the options presented select:

   “Code maturity level” options
   “Prompt for development and/or incomplete drivers”
   “Console drivers” options
   “Frame-buffer support”
   “Support for frame buffer devices (EXPERIMENTAL)”
   “EPSON LCD/CRT/TV controller support”
   “EPSON S1D13708 Support”
   “Advanced low-level driver options”
   “xbpp packed pixels support” *

   * where x is the color depth being compile for.

   If you are using the Epson PCI evaluation board then you must also select:

   “Epson PCI Bridge adapter support”

   Once you have configured the kernel options, save and exit the configuration utility.

6. Compile and install the kernel

   Build the kernel with the following sequence of commands:

   ```
   make dep
   make clean
   make bzImage
   /sbin/lilo (if running lilo)
   ```

7. Boot to the Linux operating system

   If you are using lilo (Linux Loader), modify the lilo configuration file as discussed in the kernel build README file. If there were no errors during the build, from the command prompt run:

   ```
   lilo
   ```

   and reboot your system.

Note

In order to use the S1D13708 console driver with X server, you need to configure the X server to use the FBDEV device. A good place to look for the necessary files and instructions on this process is on the Internet at www.xfree86.org

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

QNX Photon v2.0 Display Driver

Document Number: X39A-E-005-01

THIS PAGE LEFT BLANK

QNX Photon v2.0 Display Driver

The Photon v2.0 display drivers for the S1D13708 Embedded Memory LCD Controller are intended as “reference” source code for OEMs developing for QNX platforms. The driver package provides support for 8 and 16 bit-per-pixel color depths. The source code is written for portability and contains functionality for most features of the S1D13708. Source code modification is required to provide a smaller driver for mass production.

The current revision of the driver is designed for use with either QNX RTP or QNX4 from the latest product CD (Dec. 99).

The Photon v2.0 display driver is designed around a common configuration include file called **S1D13708.h,** which is generated by the configuration utility 13708CFG. This design allows for easy customization of display type, clocks, decode addresses, rotation, etc. by OEMs. For further information on 13708CFG, see the *13708CFG Configuration Program User Manual*, document number X39A-B-001-xx.

Note

The QNX display drivers are provided as “reference” source code only. They are intended to provide a basis for OEMs to develop their own drivers for QNX Photon v2.0.

This document and the source code for the QNX display drivers are updated as appropriate. Please check the Epson Research and Development website at www.erd.epson.com for the latest revisions before beginning any development.

We appreciate your comments on our documentation. Please contact us via e-mail at documentation@erd.epson.com.

Building the Photon v2.0 Display Driver

The following steps build the Photon v2.0 display driver and integrate it into the QNX operating system. These instructions assume the QNX developer environment is correctly installed and the developer is familiar with building for the QNX operating system.

Unpack the Graphics Driver Development Kit Archive

1. Install the QNX ddk package using the Package Manager utility.

   For information about the Drivers Development Kit contact QNX directly.

2. Once the ddk package is installed, copy the directory tree /usr/scr/gddk_v1.0 into the Project directory.

3. Change directory to Project/gddk_1.0/devg.

4. Unpack the display driver files using the commands:

   #gunzip S1D13708.tar.gz

   #tar –xvf S1D13708.tar

   This unpacks the files into the directory Project/gddk_1.0/devg/S1D13708.

Configure the Driver

The files **s1d13708_16.h** and **s1d13708_8.h** contain register values required to set the screen resolution, color depth (bpp), display type, rotation, etc. The **s1d13708.h** file included with the drivers may not contain applicable values and must be regenerated. The configuration program 13708CFG can be used to build new **s1d13708_16.h** and **s1d13708_8.h** files.

Note

S1d13708.h should be created using the configuration utility 13708CFG. For more information on 13708CFG, see the *13708CFG Configuration Program User Manual*, document number X39A-B-001-xx available at www.erd.epson.com.

Build the Driver

The first time the driver is built, the following command ensures that all drivers and required libraries are built. At the root of the Project source tree, type **make**.

Note

To build drivers for X86 NTO type ‘OSLIST=nto CPULIST=x86 make’.

Further builds do not require all libraries to be re-built. To build only the S1D13708 display driver, change to the directory gddk_1.0/devg/s1d13708 and type **make**.

Installing the Driver

The build step produces two library images:

- lib/disputil/nto/x86/libdisputil.so
- lib/disputil/nto/x86/libffb.so

For the loader to locate them, the files need to be renamed and copied to the lib directory.

1. Rename libdisputil.so to libdisputil.so.1 and libffb.so to libffb.so.1.
2. Copy the files new files libdisputil.so.1 and libffb.so.1 to the directory /usr/lib.
3. Copy the file devg-S1D13708.so to the /lib/dll directory.

Note
To locate the file devg-S1D13708.so, watch the output of the 'true' command during the makefile build.

4. Modify the trap file graphics-modes in the /etc/system/config directory by inserting the following lines at the top of the file.

   io-graphics -dldevg-S1D13708.so -g640x480x8 -I0 -d0x0,0x0;#640,480,8 Epson

   io-graphics -dldevg-S1D13708.so -g640x480x16 -I0 -d0x0,0x0;#640,480,16 Epson

Run the Driver

It is recommended that the driver be verified **before starting QNX with the S1D13708 as the primary display**. To verify the driver, type the following command at the root of the Project source tree (gddk_1.0 directory).

util/bench/nto/x86/o/devg-bench -dldevg/S1D13708/nto/x86/dll/devg-S1D13708.so -m**W**,**H**,**C**,**F** -d0x0,0x0

Where:

W is the configured width of the display
H is the configured height of the display
C is the color depth in bpp (either 8 or 16)
F is the configured frame rate

This command starts the bench utility which will initialize the driver as the secondary display and exercise the drivers main functions. If the display appears satisfactory, restart QNX Photon and the restart will result in the S1D13708 display driver becoming the primary display device.

Comments

- To restore the display driver to the default, comment out changes made to the trap file crt.$NODE.

S1D13708 Embedded Memory LCD Controller

Windows® CE 3.x Display Drivers

Document Number: X39A-E-006-01

THIS PAGE LEFT BLANK

WINDOWS® CE 3.0 DISPLAY DRIVER

Windows CE v3.0 display driver for the S1D13708 Embedded Memory LCD Controller is intended as “reference” source code for OEMs developing for the Microsoft Window CE platform. The driver supports 4, 8 and 16 bit-per-pixel color depths in landscape mode (no rotation), and 8 and 16 bit-per-pixel color depths in SwivelView™ 90 degree, 180 degree and 270 degree modes. The source code is written for portability and contains functionality for most features of the S1D13708. Source code modification is required to provide a smaller driver for mass production.

The Windows CE v3.0 display driver is designed around a common configuration include file called **mode0.h,** which is generated by the configuration utility 13708CFG. This design allows for easy customization of display type, clocks, decode addresses, rotation, etc. by OEMs. For further information on 13708CFG, see the *13708CFG Configuration Program User Manual*, document number X39A-B-001-xx.

Note

The Windows CE display driver is provided as “reference” source code only. They are intended to provide a basis for OEMs to develop their own drivers for Microsoft Windows CE v3.0.

This document and the source code for the Windows CE v3.0 driver is updated as appropriate. Before beginning any development, please check the Epson Research and Development Website at www.erd.epson.com for the latest revisions.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

Example Driver Builds

The following sections describes how to build the Windows CE display driver for Windows CE Platform Builder 3.00 using the GUI interface.

Build for CEPC (X86) on Windows CE Platform Builder 3.00 using the GUI Interface

1. Install Microsoft Windows 2000 Professional, or Windows NT Workstation version 4.0 with Service Pack 5 or later.
2. Install Platform Builder 3.00.
3. Start Platform Builder by double-clicking on the Microsoft Windows CE Platform Builder icon.
4. Create a new project.
   a. Select File | New.
   b. In the dialog box, select the Platforms tab.
   c. In the platforms dialog box, select “WCE Platform”, set a location for the project (such as x:\myproject), set the platform name (such as myplatform), and set the processor to “Win32 (WCE x86)”.
   d. Click the OK button.
   e. In the dialog box “WCE Platform - Step 1 of 2”, select CEPC.
   f. Click the Next button.
   g. In the dialog box “WCE Platform - Step 2 of 2”, select Maximum OS (Maxall).
   h. Click the Finish button.
   i. In the dialog box “New Platform Information”, click the OK button.
5. Set the active configuration to “Win32 (WCE x86) Release”.
   a. From the Build menu, select “Set Active Configuration”.
   b. Select “MYPLATFORM - Win32 (WCE x86) Release”.
   c. Click the OK button.

6. Add the environment variable DDI_S1D13708.
   a. From the Platform menu, select “Settings”.
   b. Select the “Environment” tab.
   c. In the Variable box, type “DDI_S1D13708”.
   d. In the Value box, type “1”.
   e. Click the Set button.
   f. Click the OK button.
7. Create a new directory S1D13708, under x:\wince300\platform\cepc\drivers\display, and copy the S1D13708 driver source code into this new directory.
8. Add the S1D13708 driver component.
   a. From the Platform menu, select “Insert | User Component”.
   b. Set “Files of type:” to “All Files (*.*)”.
   c. Select the file **x:\wince300\platform\cepc\drivers\display\S1D13708\sources**.
   d. Click the OK button.
   e. In the “User Component Target File” dialog box, select browse and then select the path and the file name of “sources” as in step c.
   f. Click the OK button.
9. Delete the component “ddi_flat”.
   a. In the Platform window, select the ComponentView tab.
   b. Show the tree for MYPLATFORM components by clicking on the ‘+’ sign at the root of the tree.
   c. Right-click on the ddi_flat component.
   d. Select “Delete”.
   e. From the File menu, select “Save Workspace”.

10. From the Platform window, click the ParameterView Tab. Show the tree for MY-PLATFORM Parameters by clicking the '+' sign at the root of the tree. Expand the WINCE300 tree and then click "Hardware Specific Files" and then double click "PLATFORM.BIB". Edit the file the **platform.bib** file and make the following two changes:

    a. Insert the following text after the line "IF ODO_NODISPLAY !":

```
IF CEPC_DDI_S1D13708
        ddi.dll    $(_FLATRELEASEDIR)\S1D13708.dll    NK SH
ENDIF
```

    b. Find the section shown below, and insert the lines as marked:

```
IF CEPC_DDI_FLAT !
IF DDI_S1D13708!                        ;Insert this line
IF CEPC_DDI_S3VIRGE !
IF CEPC_DDI_CT655X !
IF CEPC_DDI_VGA8BPP !
IF CEPC_DDI_S3TRIO64 !
IF CEPC_DDI_ATI !
        ddi.dll    $(_FLATRELEASEDIR)\ddi_flat.dll      NK SH
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF                                   ;Insert this line
ENDIF
```

11. Modify MODE0.H.

    The file MODE0.H (located in x:\wince300\platform\cepc\drivers\display\S1D13708) contains the register values required to set the screen resolution, color depth (bpp), display type, display rotation, etc.

    Before building the display driver, refer to the descriptions in the file MODE0.H for the default settings of the console driver. If the default does not match the configuration you are building for then **mode0.h** will have to be regenerated with the correct information.

    Use the program 13708CFG to generate the header file. For information on how to use 13708CFG, refer to the *13708CFG Configuration Program User Manual*, document number X39A-B-001-xx, available at www.erd.epson.com

    After selecting the desired configuration, choose "File->Export" and select the "C Header File for S1D13708 WinCE Driver" option. Save the new configuration as **mode0.h** in the \wince300\platform\cepc\drivers\display, replacing the original configuration file.

12. From the Platform window, click the ParameterView Tab. Show the tree for MY-PLATFORM Parameters by clicking the '+' sign at the root of the tree. Expand the WINCE300 tree and click "Hardware Specific Files", then double click "PLATFORM.REG". Edit the file **platform.reg** to match the screen resolution, color depth, and rotation information in **mode0.h**.

    For example, the display driver section of **platform.reg** should be as follows when using a 320x240 LCD panel with a color depth of 8 bpp and a SwivelView mode of 0° (landscape):

    ```
    ; Default for EPSON Display Driver
    ; 320x240 at 8 bits/pixel, LCD display, no rotation
    ; Useful Hex Values
    ; 640=0x280 480=0x1E0 320=140 240=0xF0
    [HKEY_LOCAL_MACHINE\Drivers\Display\S1D13708]
    "Width"=dword:140
    "Height"=dword:F0
    "Bpp"=dword:8
    "Rotation"=dword:0
    ```

13. From the Build menu, select "Rebuild Platform" to generate a Windows CE image file (NK.BIN) in the project directory **x:\myproject\myplatform\reldir\x86_release\nk.bin**.

Installation for CEPC Environment

Once the **nk.bin** file is built, the CEPC environment can be started by booting either from a floppy or hard drive configured with a Windows 9x operating system. The two methods are described below.

1. To start CEPC from a floppy drive:
   a. Create a bootable floppy disk.
   b. Copy **himem.sys** to the floppy disk and edit **config.sys** on the floppy disk to contain only the following line:

      device=a:\himem.sys

   c. Edit **autoexec.bat** on the floppy disk to contain the following lines:

      loadcepc /B:38400 /C:1 c:\nk.bin

   d. Search for **loadcepc.exe** in the Windows CE directories and copy **loadcepc.exe** to the bootable floppy disk.
   e. Copy **nk.bin** to c:\.
   f. Boot the system from the bootable floppy disk.
2. To start CEPC from a hard drive:
   a. Search for **loadcepc.exe** in the Windows CE directories and copy **loadcepc.exe** to C:\.
   b. Edit **config.sys** on the hard drive to contain only the following line:

      device=c:\himem.sys

   c. Edit **autoexec.bat** on the hard drive to contain the following lines:

      loadcepc /B:38400 /C:1 c:\nk.bin

   d. Copy nk.bin to c:\.
   e. Boot the system.

Configuration

There are several issues to consider when configuring the display driver. The issues cover debugging support, register initialization values and memory allocation. Each of these issues is discussed in the following sections.

Compile Switches

There are several switches, specific to the S1D13708 display driver, which affect the display driver.

The switches are added or removed from the compile switches in the file SOURCES.

WINCEVER

This option is automatically set to the numerical version of WinCE for version 2.12 or later. If the environment variable, _WINCEOSVER is not defined, then WINCEVER will default to 2.11. The S1D13708 display driver may test against this option to support different WinCE version-specific features.

EnablePreferVmem

This option enables the use of off-screen video memory. When this option is enabled, WinCE can optimize some BLT operations by using off-screen video memory to store images. You may need to disable this option if your off-screen video memory is limited.

ENABLE_CLOCK_CHIP

This option is used to enable support for the ICD2061A clock generator. This clock chip is used on the S5U13708B00C evaluation board. The S1D13708 display drivers can program the clock chip to support the frequencies required in the MODE tables.

If you are not using the S5U13708B00C evaluation adapter, you should disable this option.

EpsonMessages

This debugging option enables the display of EPSON-specific debug messages. These debug message are sent to the serial debugging port. This option should be disabled unless you are debugging the display driver, as they will significantly impact the performance of the display driver.

DEBUG_MONITOR

This option enables the use of the debug monitor. The debug monitor can be invoked when the display driver is first loaded and can be used to view registers, and perform a few debugging tasks. The debug monitor is still under development and is **UNTESTED**.

This option should remain disabled unless you are performing specific debugging tasks that require the debug monitor.

MonoPanel

This option is intended for the support of monochrome panels only.

The option causes palette colors to be grayscaled for correct display on a mono panel. For use with color panels this option should not be enabled.

DEBUG_BLT

This option enables special BLT debugging messages on the debugging serial port. This option, when enabled, will drastically impact display driver performance, and should only be used to track down failures in the BLT operations.

This option should be disabled unless doing BLT debugging.

Mode File

A second variable which will affect the finished display driver is the register configurations contained in the mode file.

The MODE tables (contained in files MODE0.H, MODE1.H, MODE2.H . . .) contain register information to control the desired display mode. The MODE tables must be generated by the configuration program 13708CFG.EXE. The display driver comes with one example MODE table:

- MODE0.H - LCD 8-bit STN color, 320x240, 8bpp, 70Hz

By default, only MODE0.H is used by the display driver. New mode tables can be created using the 13708CFG program. Edit the #include section of MODE.H to add the new mode table.

If you only support a single mode table, you do not need to add any information to the WinCE registry. If, however, you support more that one display mode, you should create registry values (see below) that will establish the initial display mode. If your display driver contains multiple mode tables, and if you do not add any registry values, the display driver will default to the first mode table in your list.

To select which display mode the display driver should use upon boot, add the following lines to your PLATFORM.REG file:

[HKEY_LOCAL_MACHINE\Drivers\Display\S1D13708]

"Width"=dword:140
"Height"=dword:F0
"Bpp"=dword:8
"Rotation"=dword:0

Note that all dword values are in hexadecimal, therefore 140h = 320 and F0h = 240. When the display driver starts, it will read these values in the registry and attempt to match a mode table against them. All values must be present and valid for a match to occur, otherwise the display driver will default to the first mode table in your list.

A WinCE desktop application (or control panel applet) can change these registry values, and the display driver will select a different mode upon warmboot. This allows the display driver to support different display configurations and/or orientations. An example application that controls these registry values will be made available upon the next release of the display driver; preliminary alpha code is available by special request.

Resource Management Issues

The Windows CE 3.0 OEM must deal with certain display driver issues relevant to Windows CE 3.0. These issues require the OEM balance factors such as: system vs. display memory utilization, video performance, and power off capabilities.

The section "Simple Display Driver Configuration" on page 13 provides a configuration which should work with most Windows CE platforms. This section is only intended as a means of getting started. Once the developer has a functional system, it is recommended to optimize the display driver configuration as described below in "Description of Windows CE Display Driver Issues".

Description of Windows CE Display Driver Issues

The following are some issues to consider when configuring the display driver to work with Windows CE:

1. When Windows CE enters the Suspend state (power-off), the LCD controller and display memory may lose power, depending on how the OEM sets up the system. If display memory loses power, all images stored in display memory are lost.

   If power-off/power-on features are required, the OEM has several options:

   - If display memory power is turned off, add code to the display driver to save any images in display memory to system memory before power-off, and add code to restore these images after power-on.

- If display memory power is turned off, instruct Windows CE to redraw all images upon power-on. Unfortunately it is not possible to instruct Windows CE to redraw any off-screen images, such as icons, slider bars, etc., so in this case the OEM must also configure the display driver to never use off-screen memory.
- Ensure that display memory never loses power.

2. Using off-screen display memory significantly improves display performance. For example, slider bars appear more smooth when using off-screen memory. To enable or disable the use of off-screen memory, edit the file: x:\wince300\platform\cepc\drivers\display\S1D13708\sources. In SOURCES, there is a line which, when uncommented, will instruct Windows CE to use off-screen display memory (if sufficient display memory is available):

   CDEFINES=$(CDEFINES) -DEnablePreferVmem

3. In the file PROJECT.REG under CE 3.0, there is a key called PORepaint (search the Windows CE directories for PROJECT.REG). PORepaint is relevant when the Suspend state is entered or exited. PORepaint can be set to 0, 1, or 2 as described below:

   a. PORepaint=0

      - This mode tells Windows CE not to save or restore display memory on suspend or resume.
      - Since display data is not saved and not repainted, this is the FASTEST mode.
      - Main display data in display memory must NOT be corrupted or lost on suspend. The memory clock must remain running.
      - Off-screen data in display memory must NOT be corrupted or lost on suspend. The memory clock must remain running.
      - This mode cannot be used if power to the display memory is turned off.

   b. PORepaint=1

      - This is the default mode for Windows CE.
      - This mode tells Windows CE to save the main display data to the system memory on suspend.
      - This mode is used if display memory power is going to be turned off when the system is suspended, and there is enough system memory to save the image.
      - Any off-screen data in display memory is LOST when suspended. Therefore off-screen memory usage must either be disabled in the display driver (i.e: EnablePreferVmem not defined in SOURCES file), or new OEM-specific code must be added to the display driver to save off-screen data to system memory when the system is suspended, and restored when resumed.
      - If off-screen data is used (provided that the OEM has provided code to save off-screen data when the system suspends), additional code must be added to the display driver's surface allocation routine to prevent the display driver from allocating the "main memory save region" in display memory. When WinCE OS attempts to allocate a buffer to save the main display data, WinCE OS marks the allocation request as preferring display memory. We believe this is incorrect. Code must be added to prevent this specific allocation from being allocated in display memory - it MUST be allocated from system memory.
      - Since the main display data is copied to system memory on suspend, and then simply copied back on resume, this mode is FAST, but not as fast as mode 0.

c. PORepaint=2

- This mode tells WinCE to not save the main display data on suspend, and causes WinCE to REPAINT the main display on resume.
- This mode is used if display memory power is going to be turned off when the system is suspended, and there is not enough system memory to save the image.
- Any off-screen data in display memory is LOST, and since there is insufficient system memory to save display data, off-screen memory usage MUST be disabled.
- When the system is resumed, WinCE instructs all running applications to repaint themselves. This is the SLOWEST of the three modes.

Simple Display Driver Configuration

The following display driver configuration should work with most platforms running Windows CE. This configuration disables the use of off-screen display memory and forces the system to redraw the main display upon power-on.

1. This step disables the use of off-screen display memory.
   Edit the file x:\wince300\platform\cepc\drivers\display\S1D13708\sources and change the line

   CDEFINES=$(CDEFINES) -DEnablePreferVmem

   to

   #CDEFINES=$(CDEFINES) -DEnablePreferVmem

2. This step causes the system to redraw the main display upon power-on. This step is only required if display memory loses power when Windows CE is shut down. If display memory is kept powered up (set the S1D13708 in powersave mode), then the display data will be maintained and this step can be skipped.

   Search for the file PROJECT.REG in your Windows CE directories, and inside PROJECT.REG find the key PORepaint. Change PORepaint as follows:

   “PORepaint”=dword:2

Comments

- The display driver is CPU independent, allowing use of the driver for several Windows CE Platform Builder supported platforms. The file **s1dflat.cpp** will require editing for the correct values of PhysicalPortAddr, PhysicalVmemAddr, etc.
- If you are running the display driver on hardware other than the S5U13708B00C evaluation board, you must ensure that your hardware provides the correct clock frequencies for CLKI, CLKI2 and XTAL. 13708CFG defaults to 50MHz for both CLKI and CLKI2, and 12MHz for XTAL.

  If you run the S1D13708 with a single clock source, make sure your clock sources for PCLK, BCLK, and MCLK are correctly set to use the correct clock input source. Also ensure that you enable the clock dividers as required for different display hardware.
- If you are using **13708cfg.exe** to produce multiple MODE tables, make sure you change the Mode Number setting for each mode table you generate. The display driver supports multiple mode tables, but only if each mode table has a unique mode number. For more information on setting the mode number, see the *13708CFG User Manual*, document number X39A-B-001-xx.
- The 13708CFG program assumes you are using the S5U13708B00C evaluation board, and defaults the Panel Power control to GPIO0. 13708CFG allows you to change the GPIO pin used to control panel power, or to disable the use of GPIO pins altogether. If this is changed from the default, your driver will no longer be able to control panel power on the S5U13708B00C evaluation board, and your panel may not be powered up correctly.
- At this time, the driver has been tested on the x86 CPUs and have been run with Platform Builder v3.00.

S1D13708 Embedded Memory LCD Controller

Power Consumption

Document Number: X39A-G-006-01

THIS PAGE LEFT BLANK

1 S1D13708 Power Consumption

The S1D13708 power consumption can be affected by many system design variables. Some of the variables to consider are:

- **Input Clock Frequency (CLKI/CLKI2/XTAL):** CLKI/CLKI2/XTAL frequency determines the LCD frame-rate, CPU performance to memory, and other functions – the higher the input clock frequency will result in higher power consumption.
- **CPU interface:** the S1D13708 current consumption depends on the BCLK frequency, data width, number of toggling pins, and other factors – using higher BCLK gives higher CPU performance but results in higher power consumption.
- **V_{DD} Voltage Level:** the voltage level affects power consumption – the higher the voltage, the higher the power consumption.
- **Display Mode:** the resolution and color depth affect power consumption – the higher the
  resolution/color depth, the higher the power consumption.
- **Internal CLK Divide:** internal registers allow the input clock to be divided before going to the internal logic blocks – the higher the divide, the lower the power consumption.

There is a power save mode in the S1D13708 where the power consumption is affected by various system design variables.

- **Clock States During The Power Save Mode:** disabling the clock oscillation during power save mode has substantial power savings.

All unused input pins are tied to ground or as specified in the *Hardware Functional Specification*, document number X39A-A-001-xx.

1.1 Conditions

To illustrate the various factors that can effect power, the following table list the power consumption for a typical Motorola DragonBall VZ interface implementation. The DragonBall clock output of 33MHz was the connected to CLKI and divided internally for BCLK, MCLK and PCLK.

The following table provides an example of some 320x240 and 160x160 panels and the effects on power consumption in specific environments. The following conditions apply:

- All tests had an appropriate LCD panel connected to the LCD outputs of the S1D13708.
- All tests were run with a static full color palette display.
- All tests were done using the DragonBall host bus interface (CLKI = 33MHz).

Table 1-1: S1D13708 Power Consumption

Test Condition *CORE V_{DD} = 1.8V and IO V_{DD} = 3.3V*					**Power Consumption (mA)**					
					S1D13708 Active		**Power Save Mode**			
Resolution	**Panel Type (Color)**	**Frame Rate**	**Clocks (MHz)**	**Color Depth**	**CORE**	**IO**	**CORE[1]**	**IO[1]**	**CORE[2]**	**IO[2]**
320x240	18-Bit HR-TFT	84	CLKI= 33MHz MCLK= BCLK= CLKI PCLK= CLKI/4	8	1.51	4.13	0.86	0.57	0.56	0.34
				4	1.33	3.79	—	—	—	—
				2	1.24	3.76	—	—	—	—
				1	1.20	3.78	—	—	—	—
	18-Bit TFT	72	CLKI= 33MHz MCLK= BCLK= CLKI PCLK= CLKI/4	8	1.52	4.81	—	—	—	—
	8-Bit STN Format 2	93	CLKI= 33MHz MCLK= BCLK= CLKI PCLK= CLKI/4	8	1.75	4.17	—	—	—	—
160x160	18-Bit HR-TFT	93	CLKI= 33MHz MCLK= BCLK= CLKI PCLK= CLKI/8	16	1.24	2.83	0.80	0.57	0.51	0.34
				8	1.10	2.66	—	—	—	—
				4	1.02	2.40	—	—	—	—
				2	0.98	2.37	—	—	—	—
				1	0.97	2.41	—	—	—	—

1. S1D13708 Power Save Mode enabled, CLKI is active.
2. S1D13708 Power Save Mode enabled, CLKI is inactive with input tied to ground.

2 Summary

Table 1-1:, “S1D13708 Power Consumption,” on page 4 shows that the S1D13708 power consumption depends on the specific implementation. Active Mode power consumption depends on the desired CPU performance and LCD frame-rate, whereas power save mode consumption depends on the CPU Interface and Input Clock state.

In a typical design environment, the S1D13708 can be configured to be an extremely power-efficient LCD Controller with high performance and flexibility.

THIS PAGE LEFT BLANK

EPSON®

S1D13708 Embedded Memory LCD Controller

Interfacing to the NEC VR4102 / VR4111 Microprocessors

Document Number: X39A-G-007-01

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13708 Embedded Memory LCD Controller and the NEC VR4102/4111 microprocessor. The NEC VR4102 and VR4111 microprocessors are specifically designed to support an external LCD controller.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Research and Development website at http://www.erd.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Interfacing to the NEC VR4102/VR4111

2.1 The NEC VR41XX System Bus

The VR-Series family of microprocessors features a high-speed synchronous system bus typical of modern microprocessors. Designed with external LCD controller support and Windows® CE based embedded consumer applications in mind, the VR4102/VR4111 offers a highly integrated solution for portable systems. This section is an overview of the operation of the CPU bus to establish interface requirements.

2.1.1 Overview

The NEC VR series microprocessor is designed around the RISC architecture developed by MIPS. The VR4102 microprocessor is designed around the 66MHz VR4100 CPU core and the VR4111 is designed around the 80/100MHz VR4110 core. These microprocessors support 64-bit processing. The CPU communicates with the Bus Control Unit (BCU) through its internal SysAD bus. The BCU in turn communicates with external devices with its ADD and DATA busses which can be dynamically sized for 16 or 32-bit operation.

The NEC VR4102/VR4111 can directly support an external LCD controller through a dedicated bus interface. Specific control signals are assigned for an external LCD controller in order to provide an easy interface to the CPU. A 16M byte block of memory is assigned for the LCD controller with its own chip select and ready signals available. Word or byte accesses are controlled by the system high byte signal (SHB#).

2.1.2 LCD Memory Access Cycles

Once an address in the LCD block of memory is placed on the external address bus (ADD[25:0]) the LCD chip select (LCDCS#) is driven low. The read enable (RD#) or write enable (WR#) signals are driven low for the appropriate cycle. LCDRDY is driven low by the S1D13708 to insert wait states into the cycle. The system high byte enable is driven low for 16-bit transfers and high for 8-bit transfers.

Figure 2-1: "NEC VR4102/VR4111 Read/Write Cycles," shows the read and write cycles to the LCD Controller Interface.

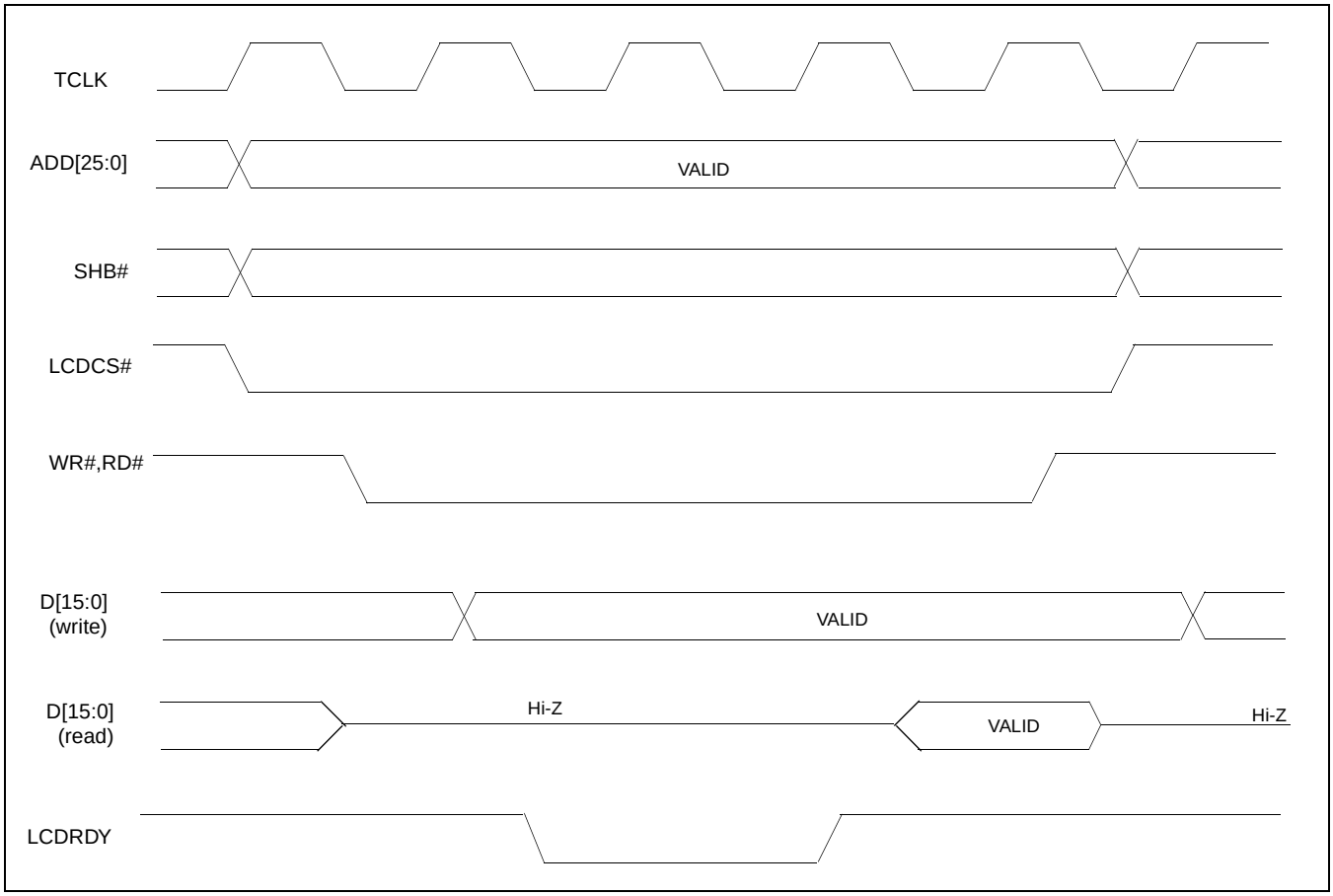


Figure 2-1: NEC VR4102/VR4111 Read/Write Cycles

3 S1D13708 Host Bus Interface

The S1D13708 directly supports multiple processors. The S1D13708 implements a 16-bit Generic #2 Host Bus Interface which is most suitable for direct connection to the NEC VR4102/4111 microprocessor. Generic #2 supports an external Chip Select, shared Read/Write Enable for high byte, and individual Read/Write Enable for low byte.

The Generic #2 Host Bus Interface is selected by the S1D13708 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13708 configuration, see Section 4.2, "S1D13708 Hardware Configuration" on page 13.

3.1 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13708 Pin Names	NEC VR4102/4111
AB[16:0]	ADD[16:0]
DB[15:0]	DAT[15:0]
WE1#	SHB#
CS#	LCDCS#
M/R#	ADD17
CLKI	BUSCLK
BS#	connect to IO V_{DD}
RD/WR#	connect to IO V_{DD}
RD#	RD#
WE0#	WR#
LCDRDY	WAIT#
RESET#	system $\overline{\text{RESET}}$

3.2 Host Bus Interface Signals

The Host Bus Interface requires the following signals:

- CLKI is a clock input which is required by the S1D13708 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For this example, BUSCLK from the NEC VR4102/4111 is used for CLKI.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the NEC VR4102/4111 address bus (ADD[16:0]) and data bus (DAT[15:0]), respectively. CNF4 must be set to select little endian mode.
- Chip Select (CS#) must be driven low by LCDCS# whenever the S1D13708 is accessed by the VR4102/4111.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address ADD17 to be connected to the M/R# line.
- WE1# connects to SHB# (the high byte enable signal from the NEC VR4102/4111) which in conjunction with address bit 0 allows byte steering of read and write operations.
- WE0# connects to WR# (the write enable signal from the NEC VR4102/4111) and must be driven low when the VR4102/4111 is writing data to the S1D13708.
- RD# connects to RD# (the read enable signal from the NEC VR4102/4111) and must be driven low when the VR4102/4111 is reading data from the S1D13708.
- WAIT# connects to LCDRDY and is a signal output from the S1D13708 that indicates the VR4102/VR4111 must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since VR4102/VR4111 accesses to the S1D13708 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13708 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The Bus Status (BS#) and Read/Write (RD/WR#) signals are not used in this implementation of the NEC VR4102/4111 interface using the Generic #2 Host Bus Interface. These pins must be tied high (connected to IO V_{DD}).

4 VR4102/VR4111 to S1D13708 Interface

4.1 Hardware Description

The NEC VR4102/VR4111 microprocessor is specifically designed to support an external LCD controller by providing the internal address decoding and control signals necessary. By using the Generic # 2 Host Bus Interface, no glue logic is required to interface the S1D13708 and the NEC VR4102/VR4111.

A pull-up resistor is attached to WAIT# to speed its rise time when terminating a cycle.

BS# (bus start) and RD/WR# are not used by the Generic #2 Host Bus Interface and must be tied high (connected to IO V_{DD}).

The following diagram shows a typical implementation of the VR4102/VR4111 to S1D13708 interface.

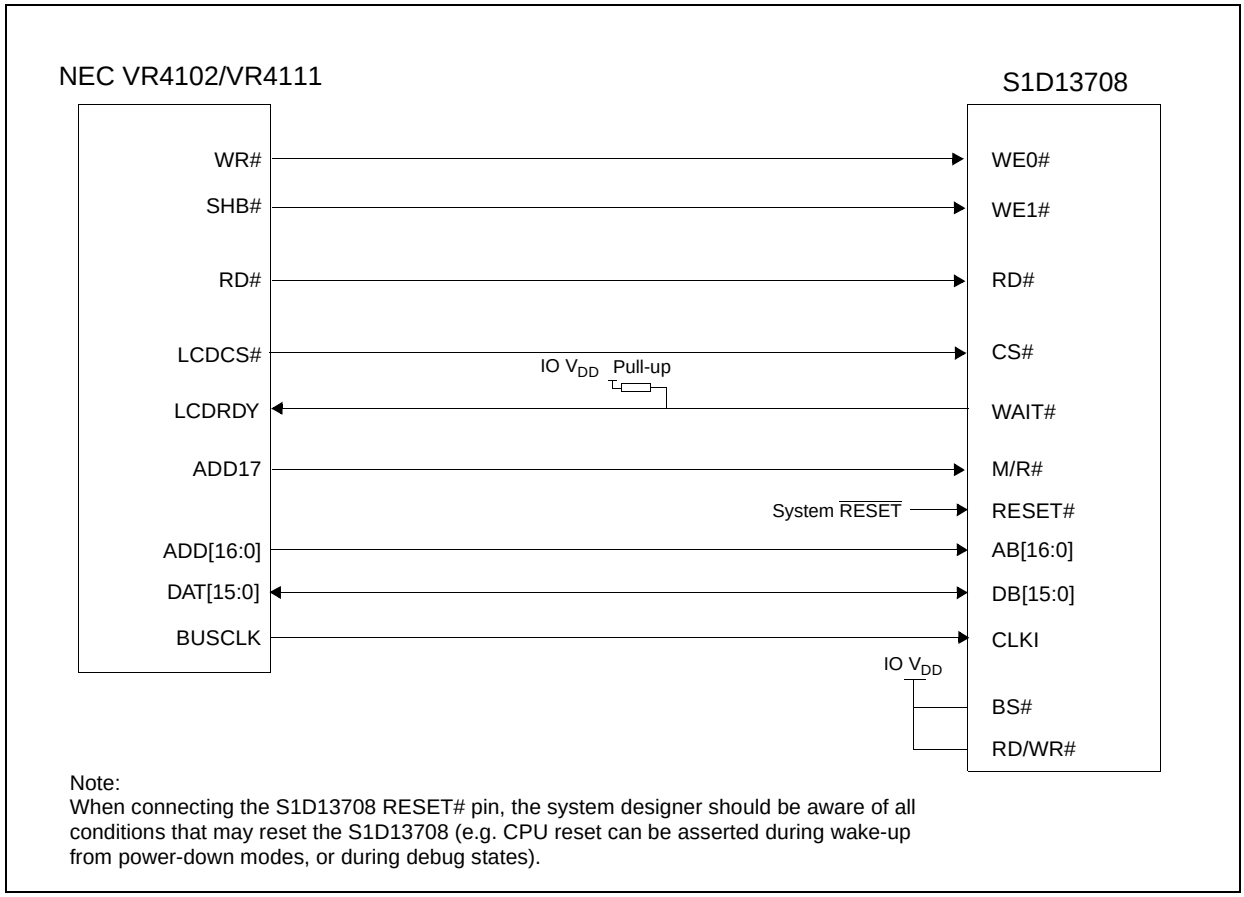


Figure 4-1: Typical Implementation of VR4102/VR4111 to S1D13708 Interface

4.2 S1D13708 Hardware Configuration

The S1D13708 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13708 to NEC VR4102/4111 interface.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13708 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[4, 2:0]	0100 = Generic #2 Little Endian Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table 4-2: "CLKI to BCLK Divide Selection" for recommended setting	

= configuration for NEC VR4102/VR4111

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for NEC VR4102/VR4111

4.3 NEC VR4102/VR4111 Configuration

The NEC VR4102/4111 provides the internal address decoding necessary to map an external LCD controller. Physical address 0A00_0000h to 0AFF_FFFFh (16M bytes) is reserved for an external LCD controller by the NEC VR4102/4111.

The S1D13708 is a memory mapped device. The S1D13708 uses two 128K byte blocks which are selected using ADD17 from the NEC VR4102/4111 (ADD17 is connected to the S1D13708 M/R# pin).The internal registers occupy the first 128K bytes block and the 80K byte display buffer occupies the second 128K byte block.

The starting address of the S1D13708 internal registers is located at 0A00_0000h and the starting address of the display buffer is located at 0A02_0000h. These blocks are aliased over the entire 16M byte address space.

Note
If aliasing is not desirable, the upper addresses must be fully decoded.

The NEC VR4102/VR4111 has a 16-bit internal register named BCUCNTREG2 located at 0B00_0002h. It must be set to the value of 0001h which indicates that LCD controller accesses use a non-inverting data bus.

The 16-bit internal register named BCUCNTREG1 (located at 0B00_0000h) must have bit D[13] (ISA/LCD bit) set to 0. This reserves 16M bytes (from 0A00_0000h to 0AFF_FFFFh) for use by the LCD controller and not as ISA bus memory space.

5 Software

Test utilities and display drivers are available for the S1D13708. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13708CFG, or by directly modifying the source. The display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13708 test utilities and display drivers are available from your sales support contact or www.erd.epson.com.

6 References

6.1 Documents

- NEC Electronics Inc., *VR4102/VR4111 64/32-bit Microprocessor Preliminary User's Manual*.
- Epson Research and Development, Inc., *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.
- Epson Research and Development, Inc., *S5U13708B00C Rev. 1.0 Evaluation Board User Manual*, document number X39A-G-004-xx.
- Epson Research and Development, Inc., *S1D13708 Programming Notes and Examples*, document number X39A-G-003-xx.

6.2 Document Sources

- NEC Electronics Inc. website: http://www.necel.com.
- Epson Research and Development Website: http://www.erd.epson.com/.

7 Technical Support

7.1 Epson LCD Controllers (S1D13708)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com/

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

7.2 NEC Electronics Inc.

NEC Electronics Inc. (U.S.A.)
Corporate Headquarters
2880 Scott Blvd.
Santa Clara, CA 95050-8062, USA
Tel: (800) 366-9782
Fax: (800) 729-9288
http://www.necel.com

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

Interfacing to the NEC VR4181A™ Microprocessor

Document Number: X39A-G-008-01

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13708 Embedded Memory LCD Controller and the NEC VR4181A microprocessor. The NEC VR4181A microprocessor is specifically designed to support an external LCD controller.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Research and Development website at http://www.erd.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Interfacing to the NEC VR4181A

2.1 The NEC VR4181A System Bus

The VR-Series family of microprocessors features a high-speed synchronous system bus typical of modern microprocessors. Designed with external LCD controller support and Windows® CE based embedded consumer applications in mind, the VR4181A offers a highly integrated solution for portable systems. This section is an overview of the operation of the CPU bus to establish interface requirements.

2.1.1 Overview

The NEC VR4181A is designed around the RISC architecture developed by MIPS. This microprocessor is designed around the 100MHz VR4110 CPU core which supports the MIPS III and MIPS16 instruction sets. The CPU communicates with external devices via an ISA interface.

While the VR4181A has an embedded LCD controller, this internal controller can be disabled to provide direct support for an external LCD controller through its external ISA bus. A 64 to 512K byte block of memory is assigned to the external LCD controller with a dedicated chip select signal (LCDCS#). Word or byte accesses are controlled by the system high byte signal (#UBE).

2.1.2 LCD Memory Access Signals

The S1D13708 requires an addressing range of 256K bytes. When the VR4181A external LCD controller chip select signal is programmed to a window of that size, the S1D13708 resides in the VR4181A physical address range of 133C 0000h to 133F FFFFh. This range is part of the external ISA memory space.

The following signals are required to access an external LCD controller. All signals obey ISA signalling rules.

- A[16:0] is the address bus.
- #UBE is the high byte enable (active low).
- #LCDCS is the chip select for the S1D13708 (active low).
- D[15:0] is the data bus.
- #MEMRD is the read command (active low).
- #MEMWR is the write command (active low).
- #MEMCS16 is the acknowledge for 16-bit peripheral capability (active low).
- IORDY is the ready signal from S1D13708.
- SYSCLK is the prescalable bus clock (optional).

Once an address in the LCD block of memory is accessed, the LCD chip select (#LCDCS) is driven low. The read or write enable signals (#MEMRD or #MEMWR) are driven low for the appropriate cycle and IORDY is driven low by the S1D13708 to insert wait states into the cycle. The high byte enable (UBE#) is driven low for 16-bit transfers and high for 8-bit transfers.

3 S1D13708 Host Bus Interface

The S1D13708 directly supports multiple processors. The S1D13708 implements a 16-bit Generic #2 Host Bus Interface which is most suitable for direct connection to the NEC VR4181A microprocessor. Generic #2 supports an external Chip Select, shared Read/Write Enable for high byte, and individual Read/Write Enable for low byte.

The Generic #2 Host Bus Interface is selected by the S1D13708 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13708 configuration, see Section 4.2, "S1D13708 Hardware Configuration" on page 13.

3.1 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13708 Pin Names	NEC VR4181A
AB[16:0]	A[16:0]
DB[15:0]	D[15:0]
WE1#	#UBE
CS#	#LCDCS
M/R#	A17
CLKI	SYSCLK
BS#	Connect to IO V_{DD}
RD/WR#	Connect to IO V_{DD}
RD#	#MEMRD
WE0#	#MEMWR
WAIT#	IORDY
RESET#	RESET#

3.2 Host Bus Interface Signals

The interface requires the following signals.

- CLKI is a clock input which is required by the S1D13708 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For this example, SYSCLK from the NEC VR4181A is used for CLKI.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the NEC VR4181A address (A[16:0]) and data bus (D[15:0]), respectively. CNF4 must be set to select little endian mode.
- Chip Select (CS#) must be driven low by #LCDCS whenever the S1D13708 is accessed by the VR4181A.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address A17 to be connected to the M/R# line.
- WE1# connects to #UBE (the high byte enable signal from the NEC VR4181A) which in conjunction with address bit 0 allows byte steering of read and write operations.
- WE0# connects to #MEMWR (the write enable signal from the NEC VR4181A) and must be driven low when the NEC VR4181A is writing data to the S1D13708.
- RD# connects to #MEMRD (the read enable signal from the NEC VR4181A) and must be driven low when the NEC VR4181A is reading data from the S1D13708.
- WAIT# connects to IORDY and is a signal which is output from the S1D13708 which indicates the NEC VR4181A must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since VR4181A accesses to the S1D13708 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13708 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The Bus Status (BS#) and Read/Write (RD/WR#) signals are not used in this implementation of the NEC VR4181A interface using the Generic #2 Host Bus Interface. These pins must be tied high (connected to IO V_{DD}).

4 VR4181A to S1D13708 Interface

4.1 Hardware Description

The NEC VR4181A microprocessor is specifically designed to support an external LCD controller by providing the internal address decoding and control signals necessary. By using the Generic # 2 Host Bus Interface, no glue logic is required to interface the S1D13708 to the NEC VR4181A.

A pull-up resistor is attached to WAIT# to speed up its rise time when terminating a cycle.

#MEMCS16 of the NEC VR4181A is connected to #LCDCS to signal that the S1D13708 is capable of 16-bit transfers.

BS# (bus start) and RD/WR# are not used by the Generic #2 Host Bus Interface and must be tied high (connected to IO V_{DD}).

The diagram below shows a typical implementation of the VR4181A to S1D13708 interface.

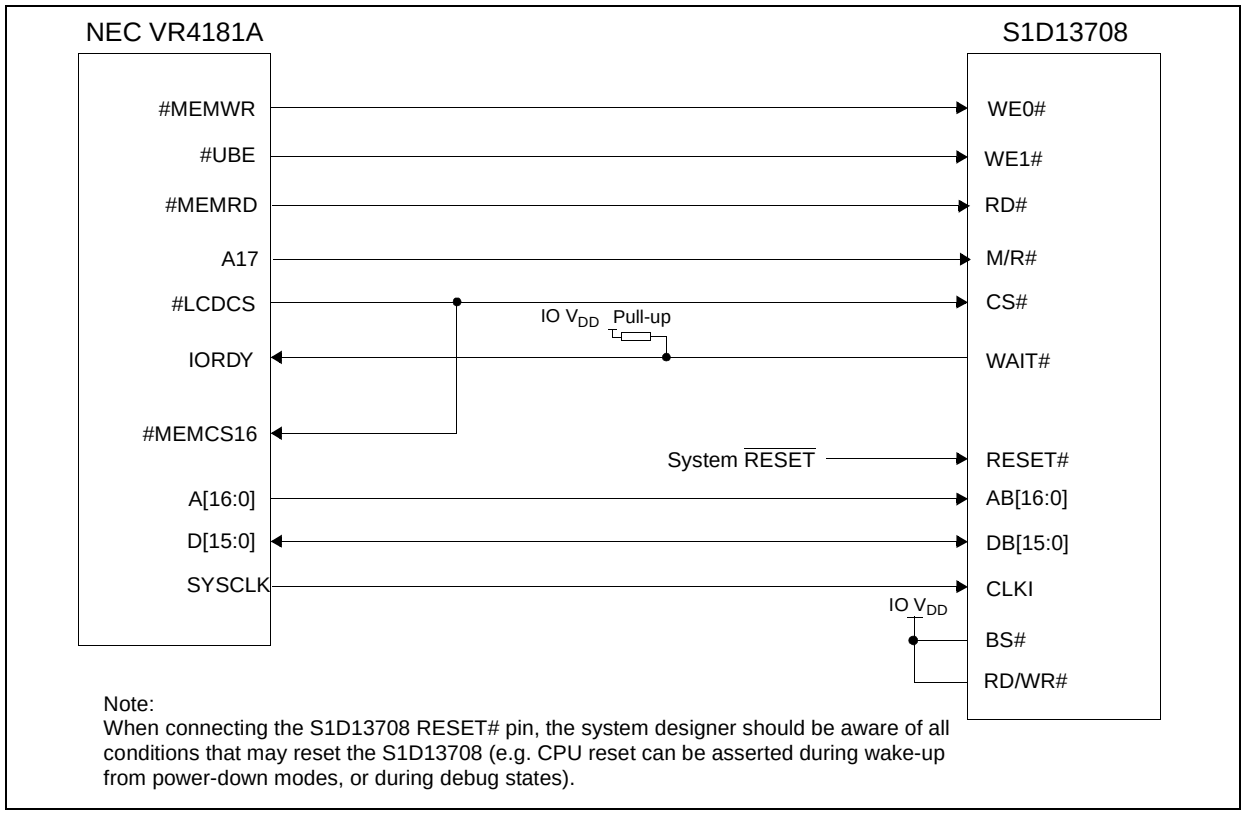


Figure 4-1: Typical Implementation of VR4181A to S1D13708 Interface

4.2 S1D13708 Hardware Configuration

The S1D13708 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13708 to NEC VR181A interface.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13708 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[4, 2:0]	0100 = Generic #2 Little Endian Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table "" for recommended setting	

= configuration for NEC VR4181A

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for NEC VR4181A

4.3 NEC VR4181A Configuration

The S1D13708 is a memory mapped device. The S1D13708 uses two 128K byte blocks which are selected using A17 from the NEC VR181A (A17 is connected to the S1D13708 M/R# pin).The internal registers occupy the first 128K bytes block and the 80K byte display buffer occupies the second 128K byte block.

When the VR4181A embedded LCD controller is disabled, the external LCD controller chip select signal (#LCDCS) decodes either a 64K byte, 128K byte, 256K byte, or 512K byte memory block in the VR4181A external ISA memory. The S1D13708 requires this block of memory to be set to 256K bytes. With this configuration, the S1D13708 internal registers starting address is located at physical memory location 133C_0000h and the display buffer is located at memory location 133E_0000h.

The NEC VR4181A must be configured through its internal registers to map the S1D13708 to the external LCD controller space. The following register values must be set.

- Register LCDGPMD at address 0B00_032Eh must be set as follows.
  - Bit 7 must be set to 1 to disable the internal LCD controller and enable the external LCD controller interface. Disabling the internal LCD controller also maps pin SHCLK to #LCDCS and pin LOCLK to #MEMCS16.
  - Bits [1:0] must be set to 10b to reserve 256Kbytes of memory address range, 133C_0000h to 133F_FFFFh for the external LCD controller.
- Register GPMD2REG at address 0B00_0304h must be set as follows.
  - Bits [9:8] (GP20MD[1:0]) must be set to 11’b to map pin GPIO20 to #UBE.
  - Bits [5:4] (GP18MD[1:0]) must be set to 01’b to map pin GPIO18 to IORDY.

5 Software

Test utilities and display drivers are available for the S1D13708. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13708CFG, or by directly modifying the source. The display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13708 test utilities and display drivers are available from your sales support contact or www.erd.epson.com.

6 References

6.1 Documents

- NEC Electronics Inc., *NEC VR4181A Target Specification*, Revision 0.5, 9/11/98
- Epson Research and Development, Inc., *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.
- Epson Research and Development, Inc., *S5U13708B00C Rev. 1.0 Evaluation Board User Manual*, document number X39A-G-004-xx.
- Epson Research and Development, Inc., *S1D13708 Programming Notes and Examples*, document number X39A-G-003-xx.

6.2 Document Sources

- NEC Electronics Inc.website: http://www.necel.com.
- Epson Research and Development Website: http://www.erd.epson.com/.

7 Technical Support

7.1 Epson LCD Controllers (S1D13708)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com/

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

7.2 NEC Electronics Inc.

NEC Electronics Inc. (U.S.A.)
Corporate Headquarters
2880 Scott Blvd.
Santa Clara, CA 95050-8062, USA
Tel: (800) 366-9782
Fax: (800) 729-9288
http://www.necel.com

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

Interfacing to the Motorola MPC821 Microprocessor

Document Number: X39A-G-009-01

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13708 Embedded Memory LCD Controller and the Motorola MPC821 microprocessor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Research and Development website at http://www.erd.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Interfacing to the MPC821

2.1 The MPC8XX System Bus

The MPC8xx family of processors feature a high-speed synchronous system bus typical of modern RISC microprocessors. This section provides an overview of the operation of the CPU bus in order to establish interface requirements.

2.2 MPC8XX Bus Overview

The MPC8xx microprocessor family uses a synchronous address and data bus. All IO is synchronous to a square-wave reference clock called MCLK (Master Clock). This clock runs at the machine cycle speed of the CPU core (typically 25 to 50 MHz). Most outputs from the processor change state on the rising edge of this clock. Similarly, most inputs to the processor are sampled on the rising edge.

Note
The external bus can run at one-half the CPU core speed using the clock control register. This is typically used when the CPU core is operated above 50 MHz.

The MPC821 can generate up to eight independent chip select outputs, each of which may be controlled by one of two types of timing generators: the General Purpose Chip Select Module (GPCM) or the User-Programmable Machine (UPM). Examples are given using the GPCM.

It should be noted that all Power PC microprocessors, including the MPC8xx family, use bit notation opposite from the convention used by most other microprocessor systems. Bit numbering for the MPC8xx always starts with zero as the most significant bit, and increments in value to the least-significant bit. For example, the most significant bits of the address bus and data bus are A0 and D0, while the least significant bits are A31 and D31.

The MPC8xx uses both a 32-bit address and data bus. A parity bit is supported for each of the four byte lanes on the data bus. Parity checking is done when data is read from external memory or peripherals, and generated by the MPC8xx bus controller on write cycles. All IO accesses are memory-mapped meaning there is no separate IO space in the Power PC architecture.

Support is provided for both on-chip (DMA controllers) and off-chip (other processors and peripheral controllers) bus masters. For further information on this topic, refer to Section 6, “References” on page 22.

The bus can support both normal and burst cycles. Burst memory cycles are used to fill on-chip cache memory, and for certain on-chip DMA operations. Normal cycles are used for all other data transfers.

2.2.1 Normal (Non-Burst) Bus Transactions

A data transfer is initiated by the bus master by placing the memory address on address lines A0 through A31 and driving $\overline{TS}$ (Transfer Start) low for one clock cycle. Several control signals are also provided with the memory address:

- TSIZ[0:1] (Transfer Size) — indicates whether the bus cycle is 8, 16, or 32-bit.
- RD/$\overline{WR}$ — set high for read cycles and low for write cycles.
- AT[0:3] (Address Type Signals) — provides more detail on the type of transfer being attempted.

When the peripheral device being accessed has completed the bus transfer, it asserts $\overline{TA}$ (Transfer Acknowledge) for one clock cycle to complete the bus transaction. Once $\overline{TA}$ has been asserted, the MPC821 will not start another bus cycle until $\overline{TA}$ has been de-asserted. The minimum length of a bus transaction is two bus clocks.

Figure 2-1: “Power PC Memory Read Cycle” illustrates a typical memory read cycle on the Power PC system bus.

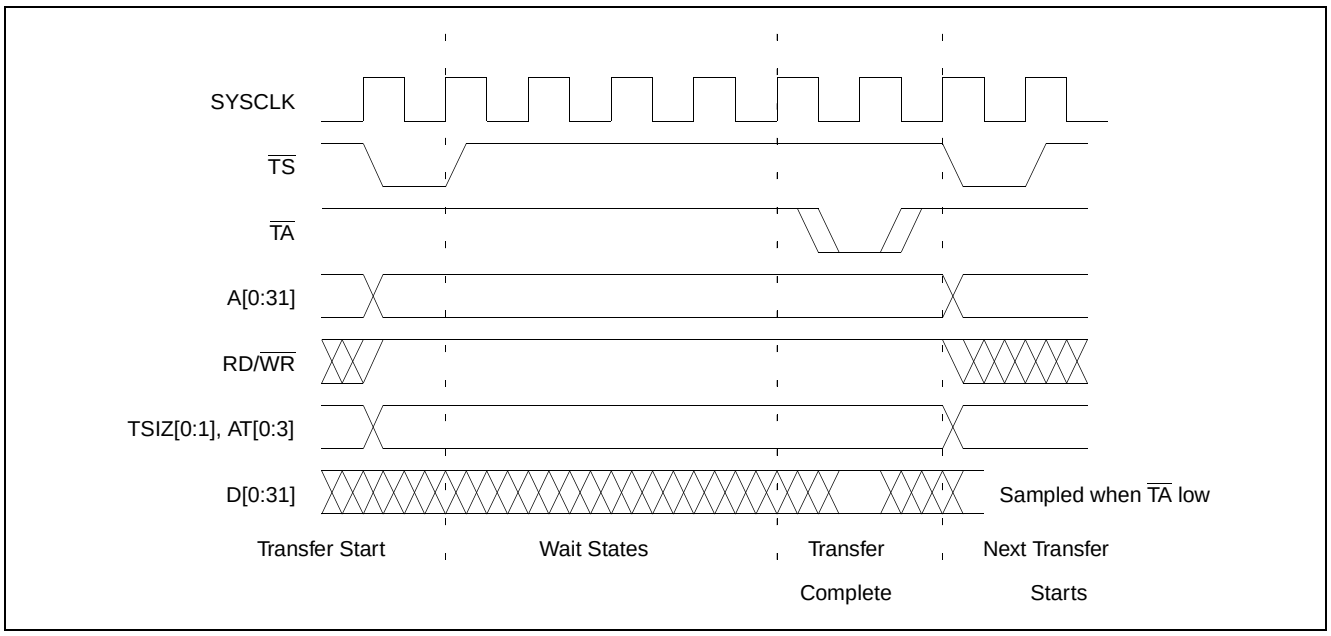


Figure 2-1: Power PC Memory Read Cycle

Figure 2-2: “Power PC Memory Write Cycle” illustrates a typical memory write cycle on the Power PC system bus.

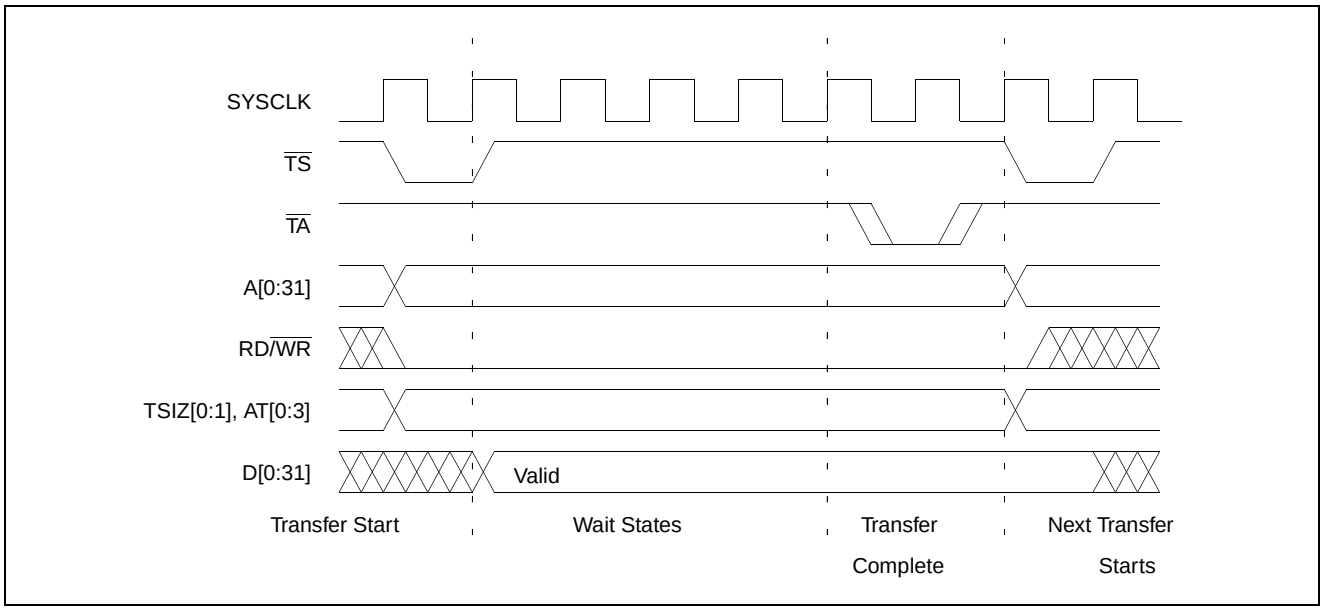


Figure 2-2: Power PC Memory Write Cycle

If an error occurs, $\overline{TEA}$ (Transfer Error Acknowledge) is asserted and the bus cycle is aborted. For example, a peripheral device may assert $\overline{TEA}$ if a parity error is detected, or the MPC821 bus controller may assert $\overline{TEA}$ if no peripheral device responds at the addressed memory location within a bus time-out period.

For 32-bit transfers, all data lines (D[0:31]) are used and the two low-order address lines A30 and A31 are ignored. For 16-bit transfers, data lines D0 through D15 are used and address line A31 is ignored. For 8-bit transfers, data lines D0 through D7 are used and all address lines (A[0:31]) are used.

Note

This assumes that the Power PC core is operating in big endian mode (typically the case for embedded systems).

2.2.2 Burst Cycles

Burst memory cycles are used to fill on-chip cache memory and to carry out certain on-chip DMA operations. They are very similar to normal bus cycles with the following exceptions:

- Always 32-bit.
- Always attempt to transfer four 32-bit words sequentially.
- Always address longword-aligned memory (i.e. A30 and A31 are always 0:0).
- Do not increment address bits A28 and A29 between successive transfers; the addressed device must increment these address bits internally.

If a peripheral is not capable of supporting burst cycles, it can assert Burst Inhibit ($\overline{BI}$) simultaneously with $\overline{TA}$, and the processor reverts to normal bus cycles for the remaining data transfers.

Burst cycles are mainly intended to facilitate cache line fills from program or data memory. They are normally not used for transfers to/from IO peripheral devices such as the S1D13708, therefore the interfaces described in this document do not attempt to support burst cycles.

2.3 Memory Controller Module

2.3.1 General-Purpose Chip Select Module (GPCM)

The General-Purpose Chip Select Module (GPCM) is used to control memory and peripheral devices which do not require special timing or address multiplexing. In addition to the chip select output, it can generate active-low Output Enable ($\overline{OE}$) and Write Enable ($\overline{WE}$) signals compatible with most memory and x86-style peripherals. The MPC821 bus controller also provides a Read/Write (RD/$\overline{WR}$) signal which is compatible with most 68K peripherals.

The GPCM is controlled by the values programmed into the Base Register (BR) and Option Register (OR) of the respective chip select. The Option Register sets the base address, the block size of the chip select, and controls the following timing parameters:

- The ACS bit field allows the chip select assertion to be delayed with respect to the address bus valid, by 0, ¼, or ½ clock cycle.
- The CSNT bit causes chip select and $\overline{WE}$ to be negated ½ clock cycle earlier than normal.
- The TRLX (relaxed timing) bit inserts an additional one clock delay between assertion of the address bus and chip select. This accommodates memory and peripherals with long setup times.
- The EHTR (Extended hold time) bit inserts an additional 1-clock delay on the first access to a chip select.
- Up to 15 wait states may be inserted, or the peripheral can terminate the bus cycle itself by asserting $\overline{TA}$ (Transfer Acknowledge).
- Any chip select may be programmed to assert $\overline{BI}$ (Burst Inhibit) automatically when its memory space is addressed by the processor core.

Figure 2-3: “GPCM Memory Devices Timing” illustrates a typical cycle for a memory mapped device using the GPCM of the Power PC.

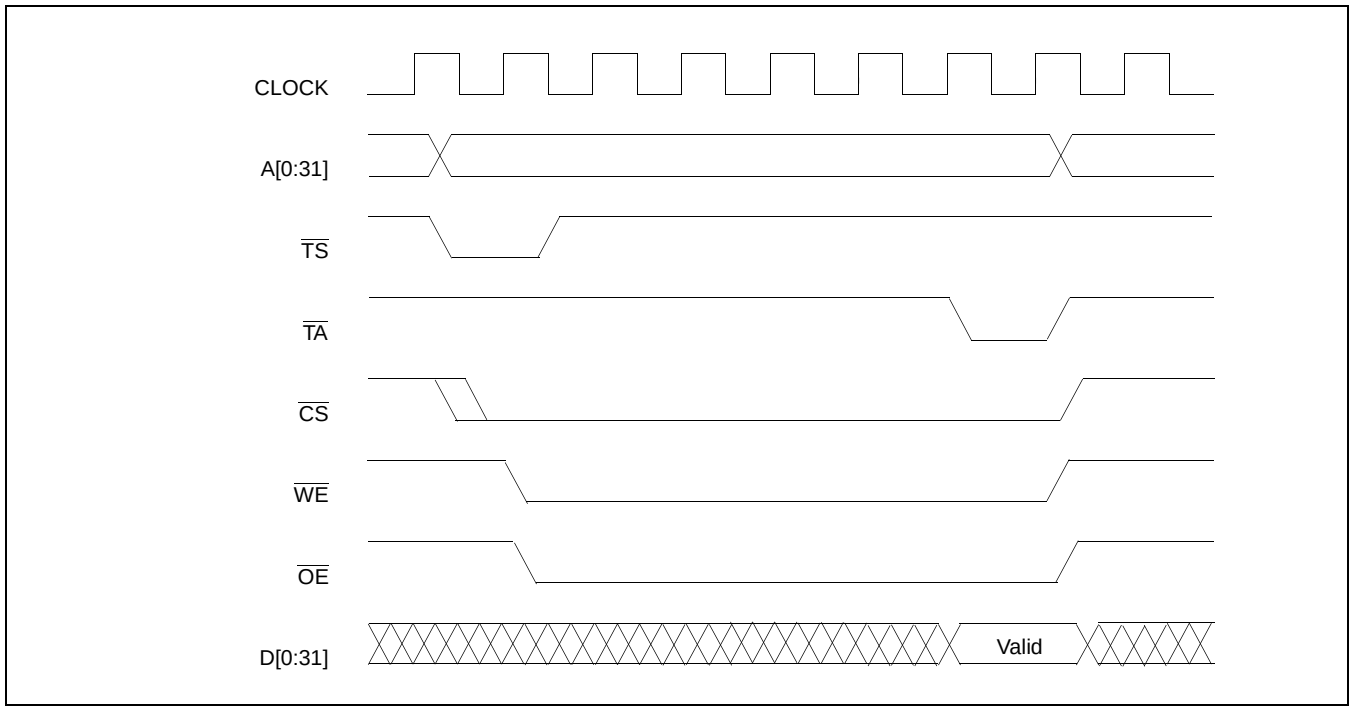


Figure 2-3: GPCM Memory Devices Timing

2.3.2 User-Programmable Machine (UPM)

The UPM is typically used to control memory types, such as Dynamic RAMs, which have complex control or address multiplexing requirements. The UPM is a general purpose RAM-based pattern generator which can control address multiplexing, wait state generation, and five general-purpose output lines on the MPC821. Up to 64 pattern locations are available, each 32 bits wide. Separate patterns may be programmed for normal accesses, burst accesses, refresh (timer) events, and exception conditions. This flexibility allows almost any type of memory or peripheral device to be accommodated by the MPC821.

In this application note, the GPCM is used instead of the UPM, since the GPCM has enough flexibility to accommodate the S1D13708 and it is desirable to leave the UPM free to handle other interfacing duties, such as EDO DRAM.

3 S1D13708 Host Bus Interface

The S1D13708 directly supports multiple processors. The S1D13708 implements a 16-bit Generic #1 Host Bus Interface which is most suitable for direct connection to the Motorola MPC821 microprocessor. Generic #1 supports a Chip Select and an individual Read Enable/Write Enable for each byte.

The Generic #1 Host Bus Interface is selected by the S1D13708 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13708 configuration, see Section 4.3, "S1D13708 Hardware Configuration" on page 18.

3.1 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13708 Pin Names	Motorola MPC821
AB[16:0]	A[15:31]
DB[15:0]	D[0:15]
WE1#	$\overline{WE0}$
CS#	$\overline{CS4}$
M/R#	A14
CLKI	SYSCLK
BS#	Connect to IO V_{DD}
RD/WR#	$\overline{OE}$ (see note)
RD#	$\overline{OE}$ (see note)
WE0#	$\overline{WE1}$
WAIT#	$\overline{TA}$
RESET#	System $\overline{RESET}$

Note

The Motorola MPC821 chip select module only handles 16-bit read cycles. As the S1D13708 uses the chip select module to generate CS#, only 16-bit read cycles are possible and both the high and low byte enables can be driven by the MPC821 signal $\overline{OE}$.

3.2 Host Bus Interface Signals

The Host Bus Interface requires the following signals.

- CLKI is a clock input which is required by the S1D13708 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For this example, SYSCLK from the Motorola MPC821 is used for CLKI.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the MPC821 address (A[15:31]) and data bus (D[0:15]), respectively. CNF4 must be set to select big endian mode.
- Chip Select (CS#) must be driven low by $\overline{CS4}$ whenever the S1D13708 is accessed by the Motorola MPC821.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address A14 to be connected to the M/R# line.
- WE0# connects to $\overline{WE1}$ (the low byte enable signal from the MPC821) and must be driven low when the MPC821 is writing the low byte to the S1D13708.
- WE1# connects to $\overline{WE0}$ (the high byte enable signal from the MPC821) and must be driven low when the MPC821 is writing the high byte to the S1D13708.
- RD# and RD/WR# are read enables for the low-order and high-order bytes, respectively. Both signals are driven low by $\overline{OE}$ when the Motorola MPC821 is reading data from the S1D13708.
- WAIT# connects to $\overline{TA}$ and is a signal which is output from the S1D13708 which indicates the MPC821 must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since MPC821 accesses to the S1D13708 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13708 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The Bus Status (BS#) signal is not used in this implementation of the MPC821 interface using the Generic #1 Host Bus Interface. This pin must be tied high (connected to IO V_{DD}).

4 MPC821 to S1D13708 Interface

4.1 Hardware Description

The interface between the S1D13708 and the MPC821 requires no external glue logic. The polarity of the WAIT# signal must be selected as active high by connecting CNF5 to IO V_{DD} (see Table 4-2:, "Summary of Power-On/Reset Configuration Options," on page 18).

BS# (bus start) is not used in this implementation and must be tied high (connected to IO V_{DD}).

The following diagram shows a typical implementation of the MPC821 to S1D13708 interface.

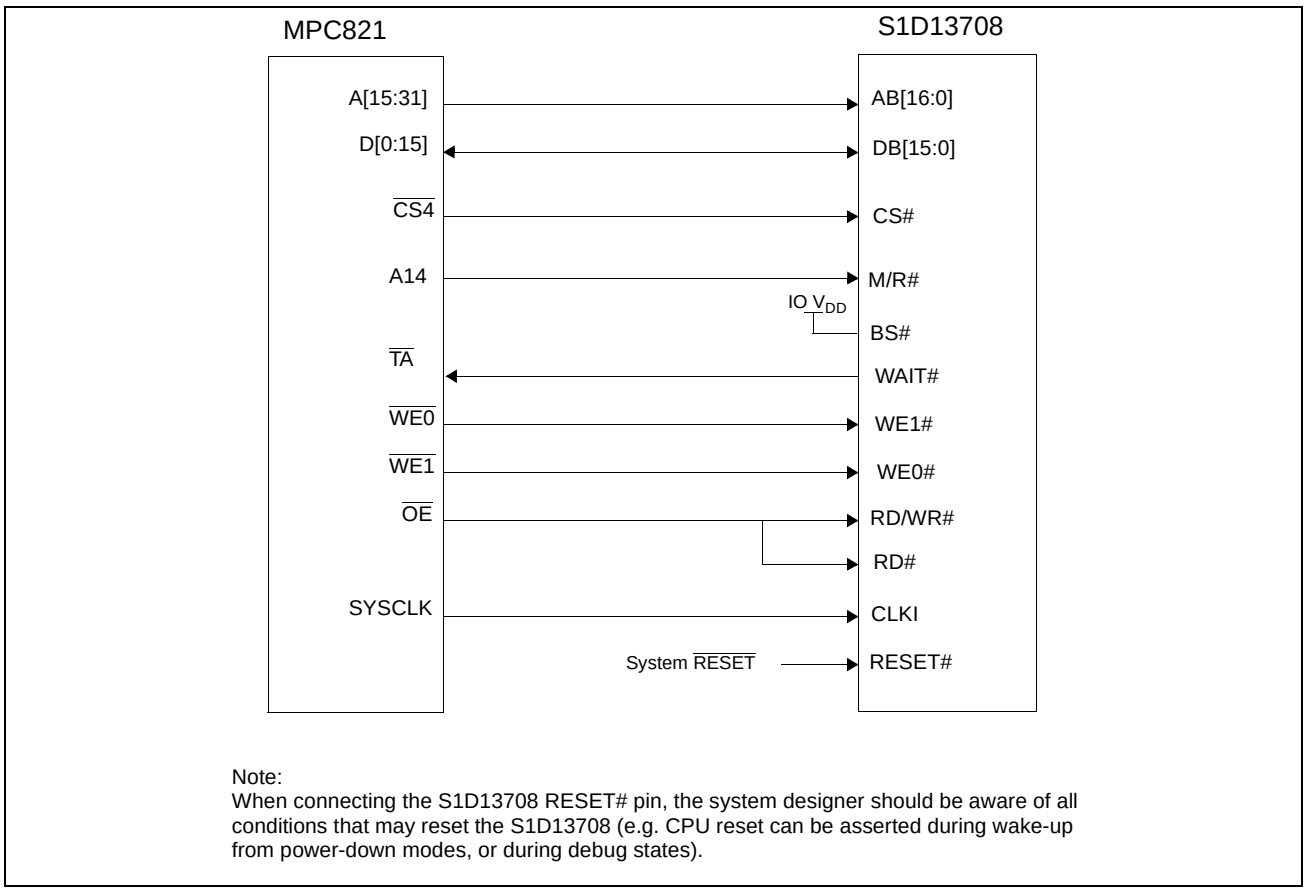


Figure 4-1: Typical Implementation of MPC821 to S1D13708 Interface

Table 4-1:, *"List of Connections from MPC821ADS to S1D13708"* on page 16 shows the connections between the pins and signals of the MPC821 and the S1D13708.

Note

The interface was designed using a Motorola MPC821 Application Development System (ADS). The ADS board has 5 volt logic connected to the data bus, so the interface included two 74F245 octal buffers on D[0:15] between the ADS and the S1D13708. In a true 3.3 volt system, no buffering is necessary.

4.2 MPC821ADS Evaluation Board Hardware Connections

The following table details the connections between the pins and signals of the MPC821 and the S1D13708.

Table 4-1: List of Connections from MPC821ADS to S1D13708

MPC821 Signal Name	MPC821ADS Connector and Pin Name	S1D13708 Signal Name
Vcc	P6-A1, P6-B1	-
A15	P6-D20	A16
A16	P6-B24	A15
A17	P6-C24	A14
A18	P6-D23	A13
A19	P6-D22	A12
A20	P6-D19	A11
A21	P6-A19	A10
A22	P6-D28	A9
A23	P6-A28	A8
A24	P6-C27	A7
A25	P6-A26	A6
A26	P6-C26	A5
A27	P6-A25	A4
A28	P6-D26	A3
A29	P6-B25	A2
A30	P6-B19	A1
A31	P6-D17	A0
D0	P12-A9	D15
D1	P12-C9	D14
D2	P12-D9	D13
D3	P12-A8	D12
D4	P12-B8	D11
D5	P12-D8	D10
D6	P12-B7	D9
D7	P12-C7	D8
D8	P12-A15	D7
D9	P12-C15	D6
D10	P12-D15	D5
D11	P12-A14	D4

Table 4-1: List of Connections from MPC821ADS to S1D13708 (Continued)

MPC821 Signal Name	MPC821ADS Connector and Pin Name	S1D13708 Signal Name
D12	P12-B14	D3
D13	P12-D14	D2
D14	P12-B13	D1
D15	P12-C13	D0
SRESET	P9-D15	RESET#
SYSCLK	P9-C2	CLKI
CS4	P6-D13	CS#
TA	P6-B6 to inverter enabled by CS#	WAIT#
WE0	P6-B15	WE1#
WE1	P6-A14	WE0#
OE	P6-B16	RD/WR#, RD#
GND	P12-A1, P12-B1, P12-A2, P12-B2, P12-A3, P12-B3, P12-A4, P12-B4, P12-A5, P12-B5, P12-A6, P12-B6, P12-A7	Vss

Note

The bit numbering of the Motorola MPC821 bus signals is reversed from the normal convention, e.g.: the most significant address bit is A0, the next is A1, A2, etc.

4.3 S1D13708 Hardware Configuration

The S1D13708 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13708 to Motorola MPC821 microprocessor.

Table 4-2: Summary of Power-On/Reset Configuration Options

S1D13708 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[4, 2:0]	1011 = Generic #1 Big Endian Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table 4-3: "CLKI to BCLK Divide Selection" for recommended settings	

= configuration for MPC821 microprocessor

Table 4-3: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for MPC821 microprocessor

4.4 Register/Memory Mapping

The DRAM on the MPC821 ADS board extends from address 0 through 3F FFFFh, so the S1D13708 is addressed starting at 40 0000h. The S1D13708 uses two 128K byte blocks which are selected using A14 from the MPC821 (A14 is connected to the S1D13708 M/R# pin). The internal registers occupy the first 128K bytes block and the 80K byte display buffer occupies the second 128K byte block.

4.5 MPC821 Chip Select Configuration

Chip select 4 is used to control the S1D13708. The following options are selected in the base address register (BR4).

- BA (0:16) = 0000 0000 0100 0000 0 – set starting address of S1D13708 to 40 0000h
- AT (0:2) = 0 – ignore address type bits.
- PS (0:1) = 1:0 – memory port size is 16 bits
- PARE = 0 – disable parity checking
- WP = 0 – disable write protect
- MS (0:1) = 0:0 – select General Purpose Chip Select module to control this chip select
- V = 1 – set valid bit to enable chip select

The following options were selected in the option register (OR4).

- AM (0:16) = 1111 1111 1100 0000 0 – mask all but upper 10 address bits; S1D13708 consumes 4M byte of address space
- ATM (0:2) = 0 – ignore address type bits
- CSNT = 0 – normal $\overline{CS}$/$\overline{WE}$ negation
- ACS (0:1) = 1:1 – delay $\overline{CS}$ assertion by ½ clock cycle from address lines
- BI = 1 – assert Burst Inhibit
- SCY (0:3) = 0 – wait state selection; this field is ignored since external transfer acknowledge is used; see SETA below
- SETA = 1 – the S1D13708 generates an external transfer acknowledge using the WAIT# line
- TRLX = 0 – normal timing
- EHTR = 0 – normal timing

4.6 Test Software

The test software to exercise this interface is very simple. It configures chip select 4 (CS4) on the MPC821 to map the S1D13708 to an unused 256K byte block of address space and loads the appropriate values into the option register for CS4. Then the software runs a tight loop reading the 13708 Revision Code Register REG[00h]. This allows monitoring of the bus timing on a logic analyzer.

The following source code was entered into the memory of the MPC821ADS using the line-by-line assembler in MPC8BUG (the debugger provided with the ADS board). Once the program was executed on the ADS, a logic analyzer was used to verify operation of the interface hardware.

It is important to note that when the MPC821 comes out of reset, its on-chip caches and MMU are disabled. If the data cache is enabled, then the MMU must be set up so that the S1D13708 memory block is tagged as non-cacheable, to ensure that accesses to the S1D13708 occurs in proper order, and also to ensure that the MPC821 does not attempt to cache any data read from or written to the S1D13708 or its display buffer.

The source code for this test routine is as follows:

```
BR4           equ         $120              ; CS4 base register
OR4           equ         $124              ; CS4 option register
MemStart      equ         $42 0000          ; address of S1D13708 display buffer
RevCodeReg    equ         $40 0000          ; address of Revision Code Register

Start         mfspr       r1,IMMR           ; get base address of internal registers
              andis.      r1,r1,$ffff       ; clear lower 16 bits to 0
              andis.      r2,r0,0           ; clear r2
              oris        r2,r2,MemStart    ; write base address
              ori         r2,r2,$0801       ; port size 16 bits; select GPCM; enable
              stw         r2,BR4(r1)        ; write value to base register
              andis.      r2,r0,0           ; clear r2
              oris        r2,r2,$ffc0       ; address mask – use upper 10 bits
              ori         r2,r2,$0708       ; normal CS negation; delay CS ½ clock;
                                            ; inhibit burst
              stw         r2,OR4(r1)        ; write to option register
              andis.      r1,r0,0           ; clear r1
              oris        r1,r1,MemStart    ; point r1 to start of S1D13708 mem space
Loop          lbz         r0,RevCodeReg(r1) ; read revision code into r1
              b           Loop              ; branch forever

end
```

Note

MPC8BUG does not support comments or symbolic equates. These have been added for clarity only.

5 Software

Test utilities and display drivers are available for the S1D13708. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13708CFG, or by directly modifying the source. The display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13708 test utilities and display drivers are available from your sales support contact or www.erd.epson.com.

6 References

6.1 Documents

- Motorola Inc., *Power PC MPC821 Portable Systems Microprocessor User's Manual*, Motorola Publication no. MPC821UM/; available on the Internet at http://www.motorola.com/.
- Epson Research and Development, Inc., *S1D13708 Hardware Functional Specification,* Document Number X39A-A-001-xx.
- Epson Research and Development, Inc., *S5U13708B00C Rev. 1.0 Evaluation Board User Manual,* Document Number X39A-G-004-xx.
- Epson Research and Development, Inc., *Programming Notes and Examples,* Document Number X39A-G-003-xx.

6.2 Document Sources

- Motorola Inc. Literature Distribution Center: (800) 441-2447.
- Motorola Inc. Website: http://www.motorola.com/.
- Epson Research and Development Website: http://www.erd.epson.com/.

7 Technical Support

7.1 EPSON LCD/CRT Controllers (S1D13708)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com/

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

7.2 Motorola MPC821 Processor

- Motorola Design Line, (800) 521-6274.
- Local Motorola sales office or authorized distributor.

THIS PAGE LEFT BLANK

EPSON®

S1D13708 Embedded Memory LCD Controller

Interfacing to the Motorola MCF5307 "ColdFire" Microprocessor

Document Number: X39A-G-010-01

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13708 Embedded Memory LCD Controller and the Motorola MCF5307 Processor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Research and Development website at http://www.erd.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Interfacing to the MCF5307

2.1 The MCF5307 System Bus

The MCF5200/5300 family of processors feature a high-speed synchronous system bus typical of modern microprocessors. This section is an overview of the operation of the CPU bus in order to establish interface requirements.

2.1.1 Overview

The MCF5307 microprocessor family uses a synchronous address and data bus, very similar in architecture to the MC68040 and MPC8xx. All outputs and inputs are timed with respect to a square-wave reference clock called BCLK0 (Master Clock). This clock runs at a software-selectable divisor rate from the machine cycle speed of the CPU core (typically 20 to 33 MHz). Both the address and the data bus are 32 bits in width. All IO accesses are memory-mapped; there is no separate IO space in the Coldfire architecture.

The bus can support two types of cycles, normal and burst. Burst memory cycles are used to fill on-chip cache memories, and for certain on-chip DMA operations. Normal cycles are used for all other data transfers.

2.1.2 Normal (Non-Burst) Bus Transactions

A data transfer is initiated by the bus master by placing the memory address on address lines A31 through A0 and driving $\overline{TS}$ (Transfer Start) low for one clock cycle. Several control signals are also provided with the memory address:

- BWE[1:0] (Transfer Size) — indicates whether the bus cycle is 8, 16, or 32-bit.
- R/$\overline{W}$ — set high for read cycles and low for write cycles.
- TT[1:0] (Transfer Type Signals) — provides more detail on the type of transfer being attempted.
- $\overline{TIP}$ (Transfer In Progress) — asserts whenever a bus cycle is active.

When the peripheral device being accessed has completed the bus transfer, it asserts $\overline{TA}$ (Transfer Acknowledge) for one clock cycle to complete the bus transaction. Once $\overline{TA}$ has been asserted, the MCF5307 will not start another bus cycle until $\overline{TA}$ has been de-asserted. The minimum length of a bus transaction is two bus clocks.

Figure 2-1: “MCF5307 Memory Read Cycle,” illustrates a typical memory read cycle on the MCF5307 system bus.

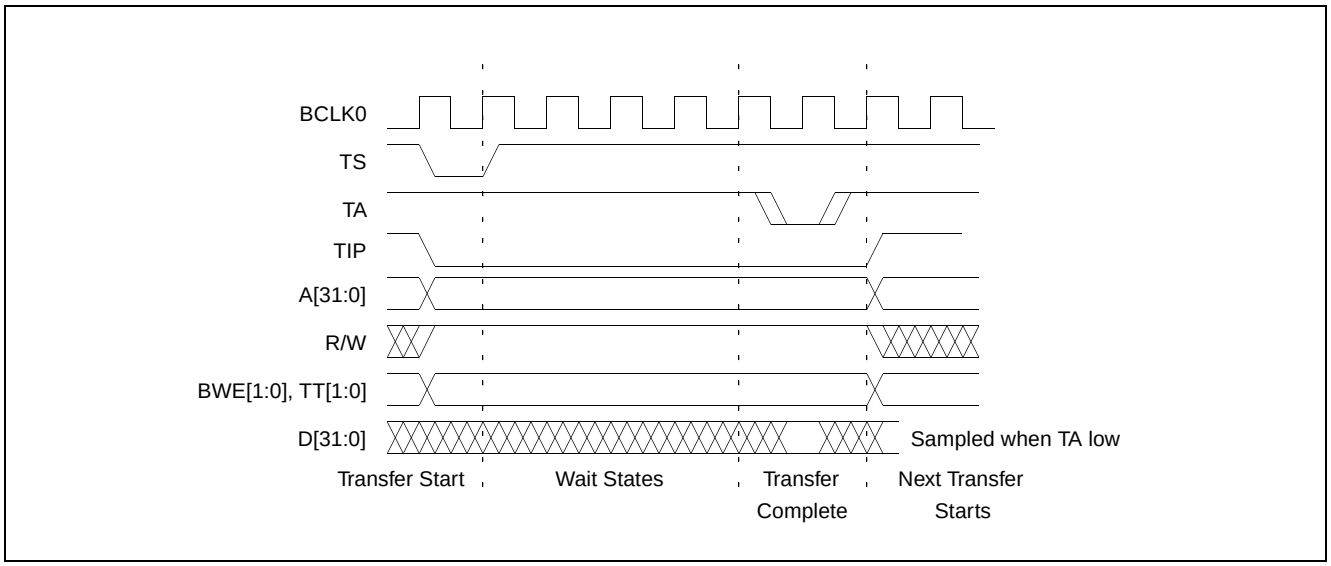


Figure 2-1: MCF5307 Memory Read Cycle

Figure 2-2: “MCF5307 Memory Write Cycle,” illustrates a typical memory write cycle on the MCF5307 system bus.

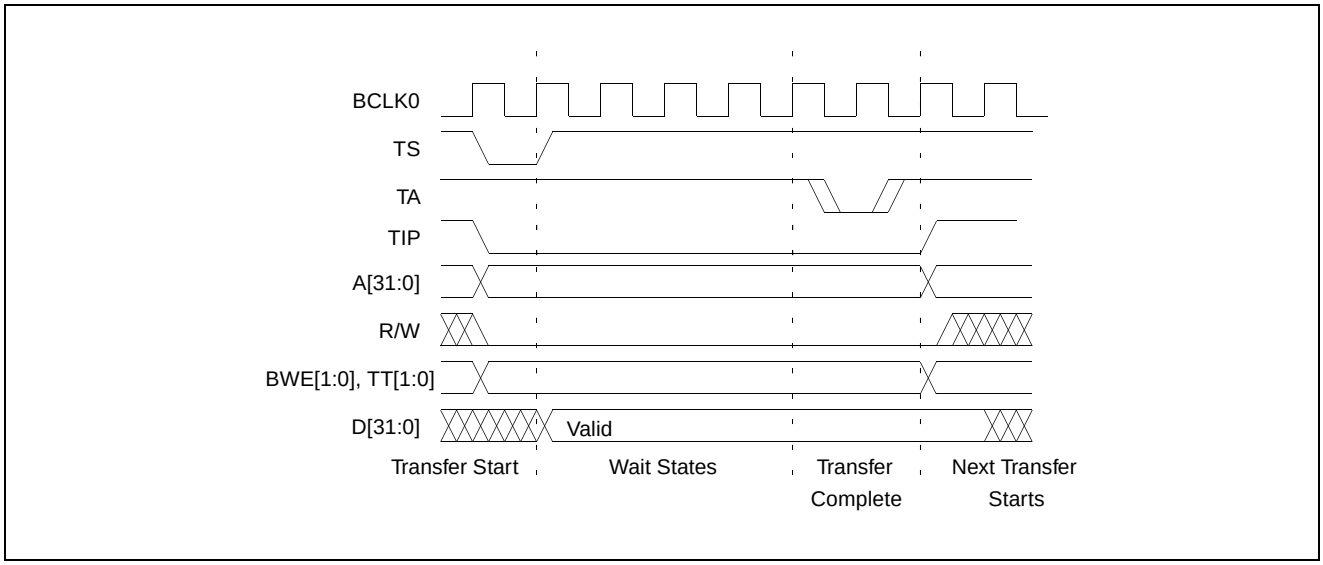


Figure 2-2: MCF5307 Memory Write Cycle

2.1.3 Burst Cycles

Burst cycles are very similar to normal cycles, except that they occur as a series of four back-to-back, 32-bit memory reads or writes. The $\overline{\text{TIP}}$ (Transfer In Progress) output is asserted continuously through the burst. Burst memory cycles are mainly intended to fill

caches from program or data memory. They are typically not used for transfers to or from IO peripheral devices such as the S1D13708. The MCF5307 chip selects provide a mechanism to disable burst accesses for peripheral devices which are not burst capable.

2.2 Chip-Select Module

In addition to generating eight independent chip-select outputs, the MCF5307 Chip Select Module can generate active-low Output Enable ($\overline{OE}$) and Write Enable ($\overline{BWE}$) signals compatible with most memory and x86-style peripherals. The MCF5307 bus controller also provides a Read/Write (R/$\overline{W}$) signal which is compatible with most 68K peripherals.

Chip selects 0 and 1 can be programmed independently to respond to any base address and block size. Chip select 0 can be active immediately after reset, and is typically used to control a boot ROM. Chip select 1 is likewise typically used to control a large static or dynamic RAM block.

Chip selects 2 through 7 have fixed block sizes of 2M bytes each. Each has a unique, fixed offset from a common, programmable starting address. These chip selects are well-suited to typical IO addressing requirements.

Each chip select may be individually programmed for:

- port size (8/16/32-bit).
- up to 15 wait states or external acknowledge.
- address space type.
- burst or non-burst cycle support.
- write protect.

Figure 2-3: “Chip Select Module Outputs Timing” illustrates a typical cycle for a memory mapped device using the GPCM of the Power PC.

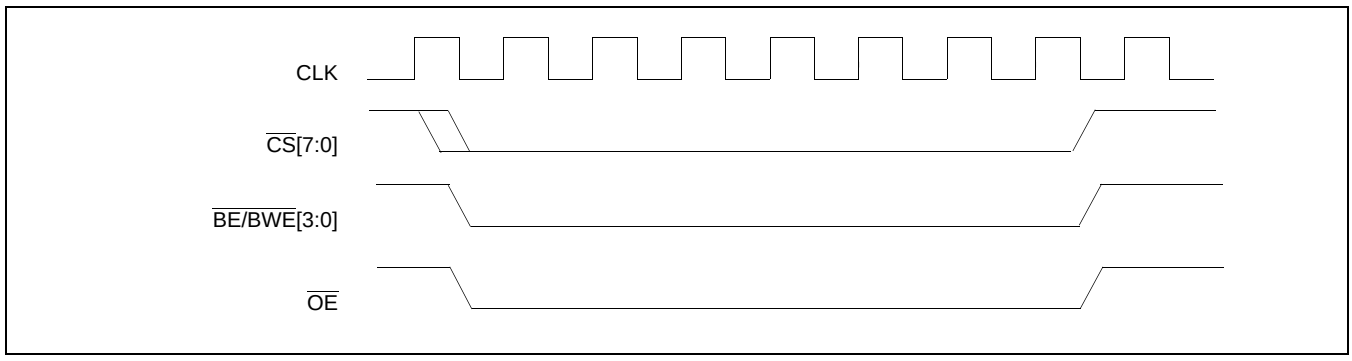


Figure 2-3: Chip Select Module Outputs Timing

3 S1D13708 Host Bus Interface

The S1D13708 directly supports multiple processors. The S1D13708 implements a 16-bit Generic #1 Host Bus Interface which is most suitable for direct connection to the Motorola MFC5307 microprocessor. Generic #1 supports a Chip Select and an individual Read Enable/Write Enable for each byte.

The Generic #1 Host Bus Interface is selected by the S1D13708 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13708 configuration, see Section 4.2, “S1D13708 Hardware Configuration” on page 14.

The S1D13708 clock (CLKI) is taken from the system host bus. The system clock source will drive all required internal clocks. If they are not used, the CLKI2 and XTAL inputs should be tied to ground.

3.1 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13708 Pin Names	Motorola MCF5307
AB[16:0]	A[16:0]
DB[15:0]	D[31:16]
WE1#	$\overline{\text{BWE1}}$
CS#	$\overline{\text{CS4}}$
M/R#	A17
CLKI	BCLK0
BS#	Connect to IO V_{DD}
RD/WR#	$\overline{\text{OE}}$
RD#	$\overline{\text{OE}}$
WE0#	$\overline{\text{BWE0}}$
WAIT#	$\overline{\text{TA}}$
RESET#	system $\overline{\text{RESET}}$

3.2 Host Bus Interface Signals

The Host Bus Interface requires the following signals.

- CLKI is a clock input which is required by the S1D13708 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For this example, BCLK0 from the Motorola MCF5307 is used for CLKI.
- The address inputs AB[16:0] connect directly to the MCF5307 address bus (A[16:0]).
- DB[7:0] connects D[23:16] (the MCF5307 low order byte). DB[15:8] connects to D[31:24] (the MCF5307 high order byte). CNF4 must be set to select big endian mode.
- Chip Select (CS#) must be driven low by $\overline{CS4}$ whenever the S1D13708 is accessed by the Motorola MCF5307.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address A17 to be connected to the M/R# line.
- WE0# connects to $\overline{BWE0}$ (the low byte enable signal from the MCF5307) and must be driven low when the MCF5307 is writing the low byte to the S1D13708.
- WE1# connects to $\overline{BWE1}$ (the high byte enable signal from the MCF5307) and must be driven low when the MCF5307 is writing the high byte to the S1D13708.
- RD# and RD/WR# are read enables for the low-order and high-order bytes, respectively. Both signals are driven low by $\overline{OE}$ when the Motorola MCF5307 is reading data from the S1D13708.
- WAIT# connects to $\overline{TA}$ and is a signal which is output from the S1D13708 that indicates the host CPU must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since host CPU accesses to the S1D13708 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13708 internal registers and/or refresh memory. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete. This signal is active low and may need to be inverted if the host CPU wait state signal is active high.
- The Bus Status (BS#) signal is not used in the bus interface for Generic #1 mode and must be tied high to IO V_{DD}.

4 MCF5307 To S1D13708 Interface

4.1 Hardware Description

The interface between the S1D13708 and the MCF5307 requires no external glue logic. The polarity of the WAIT# signal must be selected as active high by connecting CNF5 to IO V_{DD} (see Table 4-1:, "Summary of Power-On/Reset Configuration Options," on page 14).

The following diagram shows a typical implementation of the MCF5307 to S1D13708 interface.

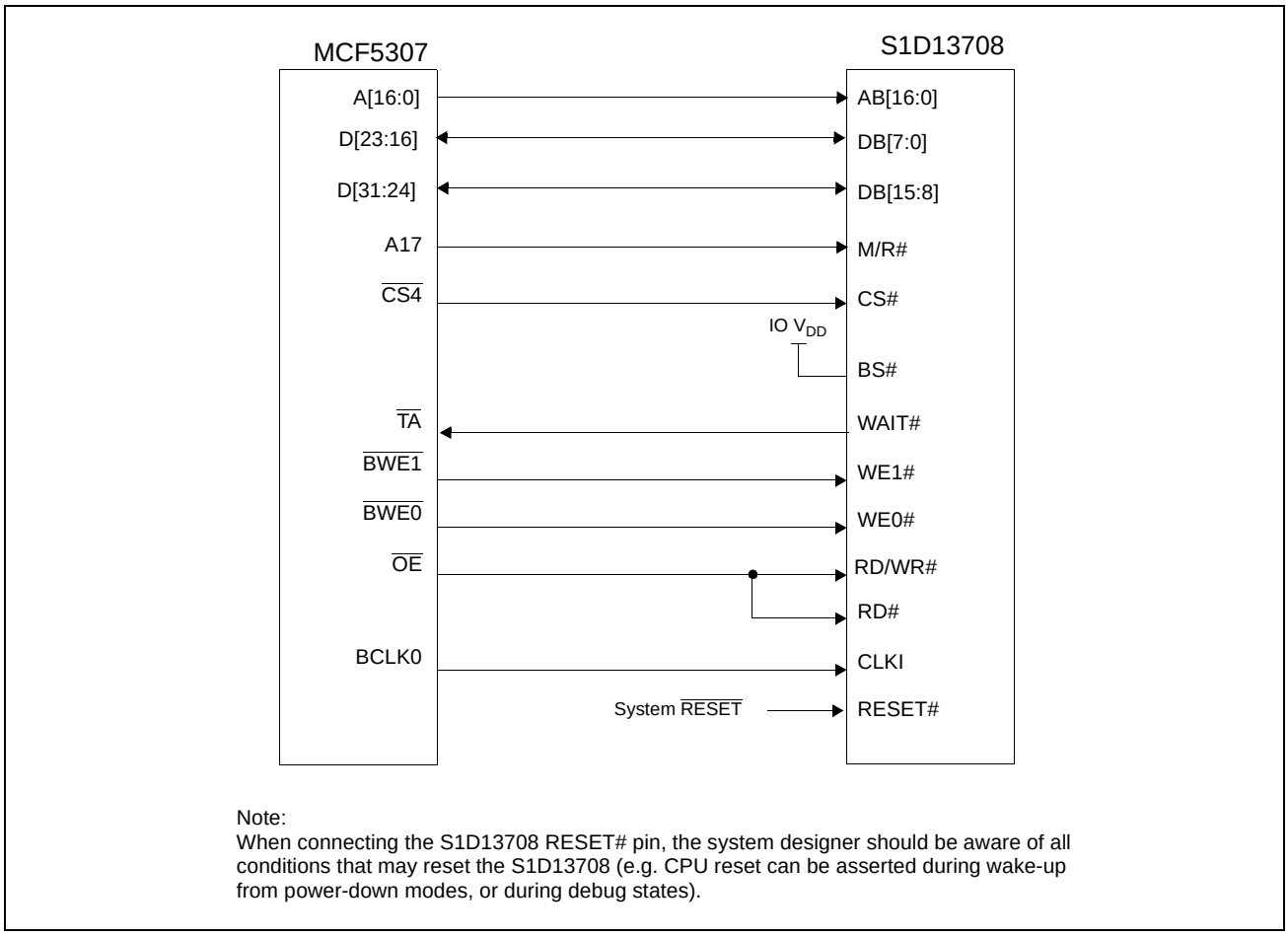


Figure 4-1: Typical Implementation of MCF5307 to S1D13708 Interface

4.2 S1D13708 Hardware Configuration

The S1D13708 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13708 to Motorola MFC5307 microprocessor.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13708 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[4, 2:0]	1011 = Generic #1 Big Endian Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	See Table 4-2: “CLKI to BCLK Divide Selection” for recommended setting	

= configuration for MFC5307 host bus interface

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for MFC5307 host bus interface

4.3 Register/Memory Mapping

The S1D13708 uses two 128K byte blocks which are selected using A17 from the MCF5307 (A17 is connected to the S1D13708 M/R# pin). The internal registers occupy the first 128K bytes block and the 80K byte display buffer occupies the second 128K byte block. These two blocks of memory are aliased over the entire 2M byte space.

Note
If aliasing is not desirable, the upper addresses must be fully decoded.

4.4 MCF5307 Chip Select Configuration

Chip Selects 0 and 1 have programmable block sizes from 64K bytes through 2G bytes. However, these chip selects would normally be needed to control system RAM and ROM. Therefore, one of the IO chip selects CS2 through CS7 is required to address the entire address space of the S1D13708. These IO chip selects have a fixed, 2M byte block size. In the example interface, chip select 4 is used to control the S1D13708. The CSBAR register should be set to the upper 8 bits of the desired base address.

The following options should be selected in the chip select mask registers (CSMR4/5).

- WP = 0 – disable write protect
- AM = 0 - enable alternate bus master access to the S1D13708
- C/I = 1 - disable CPU space access to the S1D13708
- SC = 1 - disable Supervisor Code space access to the S1D13708
- SD = 0 - enable Supervisor Data space access to the S1D13708
- UC = 1 - disable User Code space access to the S1D13708
- UD = 0 - enable User Data space access to the S1D13708
- V = 1 - global enable (“Valid”) for the chip select

The following options should be selected in the chip select control registers (CSCR4/5).

- WS0-3 = 0 - no internal wait state setting
- AA = 0 - no automatic acknowledgment
- PS (1:0) = 1:0 – memory port size is 16 bits
- BEM = 0 – Byte enable/write enable active on writes only
- BSTR = 0 – disable burst reads
- BSTW = 0 – disable burst writes

5 Software

Test utilities and display drivers are available for the S1D13708. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13708CFG, or by directly modifying the source. The display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13708 test utilities and display drivers are available from your sales support contact or www.erd.epson.com.

6 References

6.1 Documents

- Motorola Inc., *MCF5307 ColdFire® Integrated Microprocessor User's Manual*, Motorola Publication no. MCF5307UM; available on the Internet at http://www.motorola.com/.
- Epson Research and Development, Inc., *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.
- Epson Research and Development, Inc., *S5U13708B00C Rev. 1.0 Evaluation Board User Manual*, document number X39A-G-004-xx.
- Epson Research and Development, Inc., *S1D13708 Programming Notes and Examples*, document number X39A-G-003-xx.

6.2 Document Sources

- Motorola Inc. Literature Distribution Center: (800) 441-2447.
- Motorola Inc. Website: http://www.motorola.com/.
- Epson Research and Development Website: http://www.erd.epson.com/.

7 Technical Support

7.1 EPSON LCD Controllers (S1D13708)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com/

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

7.2 Motorola MCF5307 Processor

- Motorola Design Line, (800) 521-6274.
- Local Motorola sales office or authorized distributor.

EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

Connecting to the Sharp HR-TFT Panels

Document Number: X39A-G-011-01

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware environment required to connect to the Sharp HR-TFT panels directly supported by the S1D13708. These panels are:

- Sharp LQ031B1DDXX 160 x 160 HR-TFT panel.
- Sharp LQ039Q2DS01 320 x 240 HR-TFT panel.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Research and Development website at www.erd.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Connecting to the Sharp LQ039Q2DS01 HR-TFT

2.1 External Power Supplies

The S1D13708 provides all necessary data and control signals to connect to the Sharp LQ039Q2DS01 320 x 240 HR-TFT panel. However, it does not provide any of the voltages required for gray scaling, gate driving, or for the digital and analog supplies. Therefore, external supplies must be designed for any device utilizing the LQ039Q2DS01.

2.1.1 Gray Scale Voltages for Gamma Correction

The standard gray scale voltages can be generated using a precise resistor divider network that supplies two sets (A and B) of nine reference voltages to a National Semiconductor 9-Channel Buffer Amplifier (LMC6009). The LMC6009 buffers these nine reference voltages and outputs them to the panel column drivers. The A/B inputs allow the two sets of reference voltages to be alternated, compensating for asymmetrical gamma characteristics during row inversion. This input is controlled by the S1D13708 output signal REV which toggles every time a horizontal sync signal is sent to the panel.

The REV signal is also used to generate the highest gray scale voltage (V0 or black) by buffering REV and shifting its maximum level to the maximum gray scale voltage (CON_POWER). CON_POWER is supplied by a National Semiconductor micropower Voltage Regulator (LP2951). Figure 2-1: “Sharp LQ039Q2DS01 Gray Scale Voltage (V0-V9) Generation” shows the schematic for gray scale voltage generation.

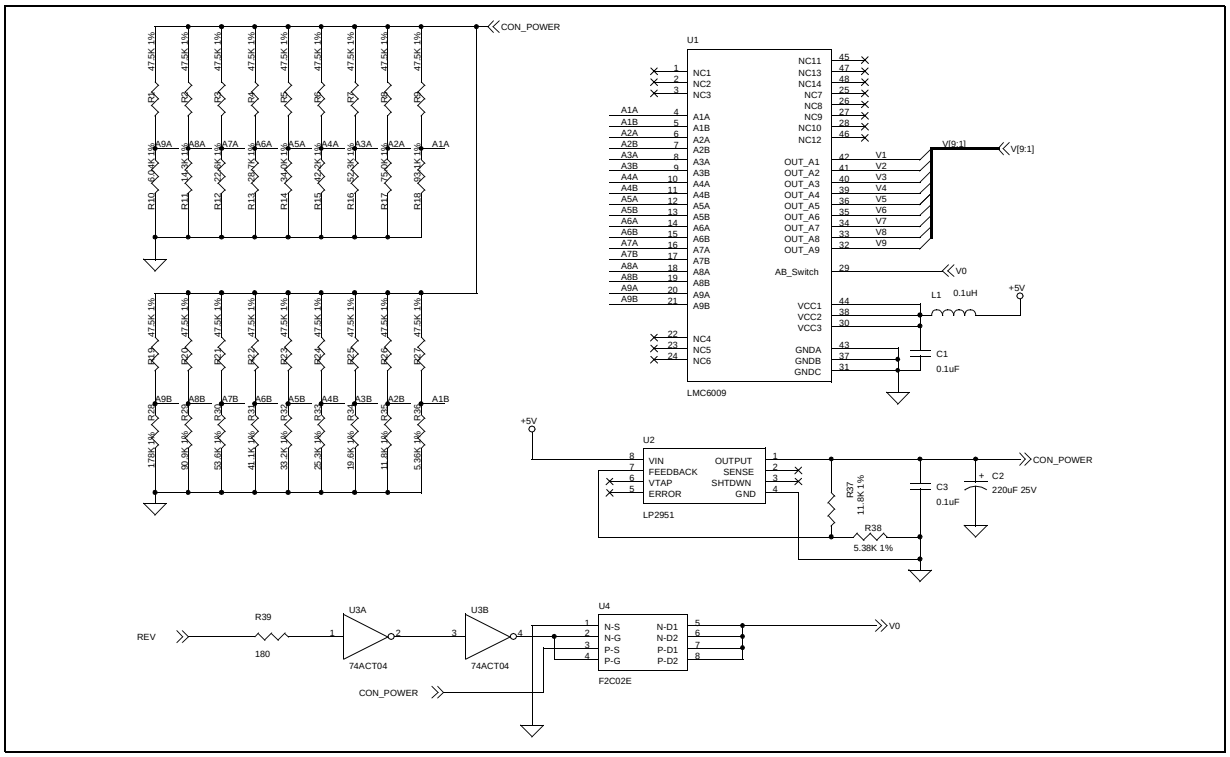


Figure 2-1: Sharp LQ039Q2DS01 Gray Scale Voltage (V0-V9) Generation

2.1.2 Digital/Analog Power Supplies

The digital power supply (VSHD) must be connected to a 3.3V supply. The analog power supply (VSHA) must be connected to a 5.0V supply.

2.1.3 DC Gate Driver Power Supplies

The gate driver high level power supply (V_{DD}) and the gate driver logic low power supply (V_{SS}) have typical values of +15V and -15V respectively. These power supplies can be provided by a Linear Technology high efficiency switching regulator (LT1172). The two power supplies can be adjusted through their allowable ranges using the potentiometer VR1.

The gate driver logic high power supply (V_{CC}) is defined as V_{SS} + VSHD. The typical V_{CC} voltage of -11.7V can be supplied from V_{SS} using a 3.3V zener diode which provides the necessary voltage change.

Figure 2-2: "Panel Gate Driver DC Power Supplies" shows the schematic for V_{SS}, V_{DD} and V_{CC}.

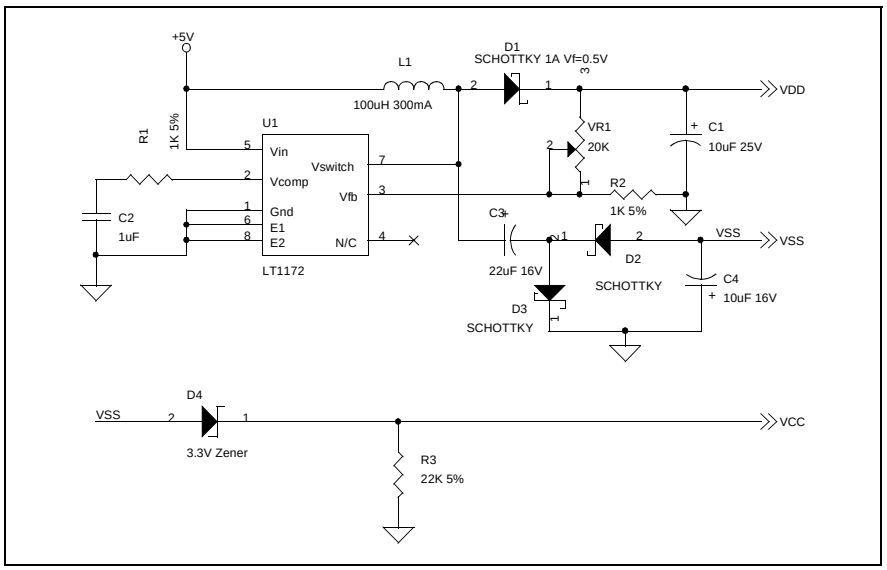


Figure 2-2: Panel Gate Driver DC Power Supplies

2.1.4 AC Gate Driver Power Supplies

The gate drive low level power supply (V_{EE}) is an AC power supply with a DC offset voltage (offset typically -9.0V). The AC component is the common electrode driving signal (V_{COM}) which has a voltage of ±2.5V. V_{COM} must be alternated every horizontal period and every vertical period. The S1D13708 output signal REV accomplishes this function and generates the alternating V_{COM} signal which is superimposed onto V_{EE}. Figure 2-3: "Panel Gate Driver AC Power Supplies," on page 10 shows the schematic for generating V_{COM} and V_{EE}.

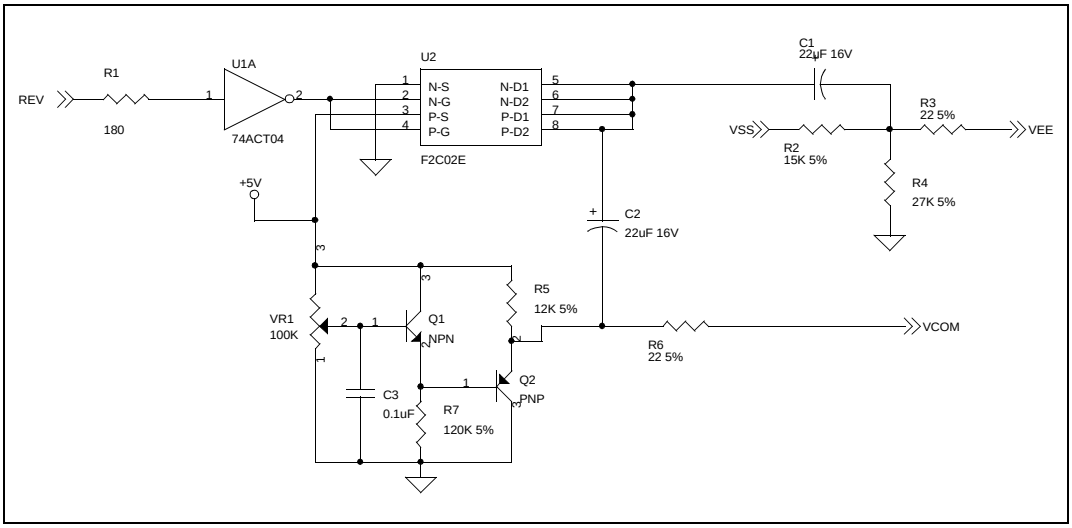
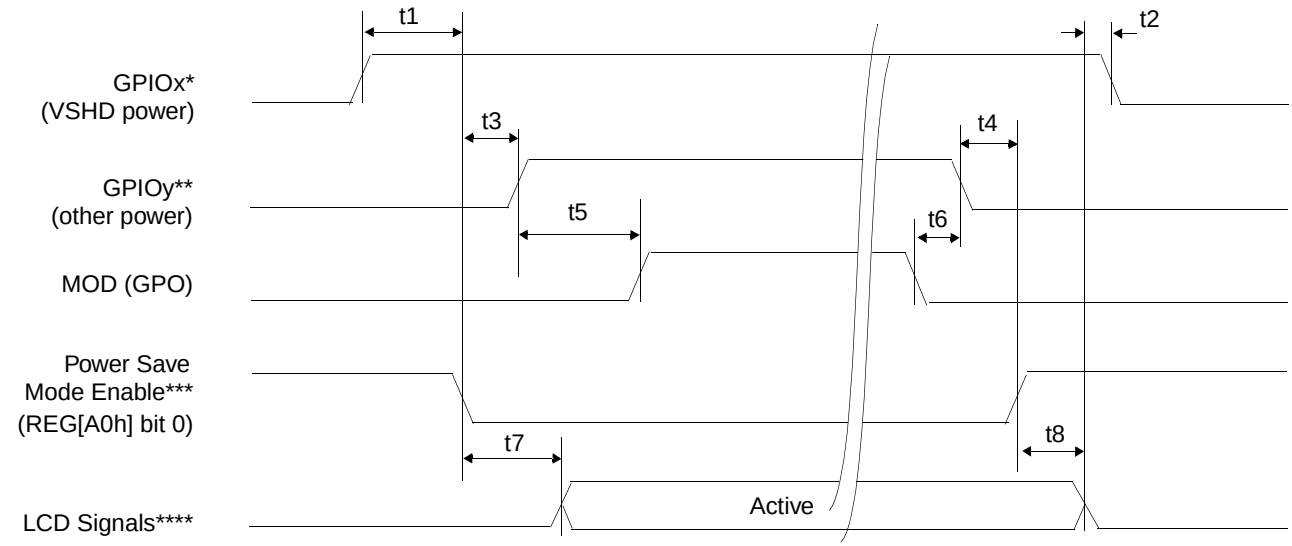


Figure 2-3: Panel Gate Driver AC Power Supplies

2.2 HR-TFT MOD Signal

The HR-TFT panel uses an input signal (MOD) to control the power-on sequencing of the panel. This HR-TFT signal should not be confused with the S1D13708 signal DRDY (referred to as MOD for passive panels).

To power-on the HR-TFT panel, MOD must be held low until the power supply has been turned on for more than two FRAMES. To power-off the HR-TFT panel, MOD must be forced low before the power supply is turned off. This sequencing requires two software controlled GPIO pins from the S1D13708 (see Figure 2-4: "HR-TFT Power-On/Off Sequence Timing").



*It is recommended to use one of the general purpose IO pins GPIO[6:4] to control the digital power supply VSHD.
** It is recommended to use one of the general purpose IO pins GPIO[6:4] to control the other power supplies required by the HR-TFT panel.
***The S1D13708 LCD power-on/off sequence is activated by programming the Power Save Mode Enable bit (REG[A0h] bit
****LCD Signals include: FPDAT[17:0], FPSHIFT, FPLINE, FPFRAME, and GPIO[3:0].

Figure 2-4: HR-TFT Power-On/Off Sequence Timing

Table 2-1: HR-TFT Power-On/Off Sequence Timing

Symbol	Parameter	Min	Max	Units
t1	LCD Power (VSHD) active to Power Save Mode disabled	0		ns
t2	LCD signals low to LCD Power (VSHD) inactive	0		ns
t3	Power Save Mode disabled to LCD Power (other) active	0		ns
t4	LCD Power (other) inactive to Power Save Mode enabled	0		ns
t5	LCD Power (other) active to MOD active	2		FRAME
t6	MOD inactive to LCD Power (other) inactive	0		ns
t7	Power Save Mode disabled to LCD signals active		20	ns
t8	Power Save Mode enabled to LCD signals low		20	ns

2.3 S1D13708 to LQ039Q2DS01 Pin Mapping

Table 2-2: S1D13708 to LQ039Q2DS01 Pin Mapping

LCD Pin No.	LCD Pin Name	S1D13708 Pin Name	Description	Remarks
1	VDD	-	Power supply of gate driver (high level)	See Section 2.1, “External Power Supplies” on page 8
2	VCC	-	Power supply of gate driver (logic high)	See Section 2.1, “External Power Supplies” on page 8
3	MOD	-	Control signal of gate driver	See Section 2.2, “HR-TFT MOD Signal” on page 11
4	MOD	-	Control signal of gate driver	See Section 2.2, “HR-TFT MOD Signal” on page 11
5	U/L	-	Selection for vertical scanning direction	Connect to VSHD (top / bottom scanning)
6	SPS	FPFRAME	Start signal of gate driver	
7	CLS	GPIO1	Clock signal of gate driver	
8	VSS	-	Power supply of gate driver (logic low)	See Section 2.1, “External Power Supplies” on page 8
9	VEE	-	Power supply of gate driver (low level)	See Section 2.1, “External Power Supplies” on page 8
10	VEE	-	Power supply of gate driver (low level)	See Section 2.1, “External Power Supplies” on page 8
11	VCOM	-	Common electrode driving signal	See Section 2.1, “External Power Supplies” on page 8
12	VCOM	-	Common electrode driving signal	See Section 2.1, “External Power Supplies” on page 8
13	SPL	GPIO3	Sampling start signal for left / right scanning	
14	R0	FPDAT11	Red data signal (LSB)	
15	R1	FPDAT10	Red data signal	
16	R2	FPDAT9	Red data signal	
17	R3	FPDAT2	Red data signal	
18	R4	FPDAT1	Red data signal	
19	R5	FPDAT0	Red data signal (MSB)	
20	G0	FPDAT14	Green data signal (LSB)	
21	G1	FPDAT13	Green data signal	
22	G2	FPDAT12	Green data signal	
23	G3	FPDAT5	Green data signal	
24	G4	FPDAT4	Green data signal	
25	G5	FPDAT3	Green data signal (MSB)	

Table 2-2: S1D13708 to LQ039Q2DS01 Pin Mapping (Continued)

LCD Pin No.	LCD Pin Name	S1D13708 Pin Name	Description	Remarks
26	B0	FPDAT17	Blue data signal (LSB)	
27	B1	FPDAT16	Blue data signal	
28	B2	FPDAT15	Blue data signal	
29	B3	FPDAT8	Blue data signal	
30	B4	FPDAT7	Blue data signal	
31	B5	FPDAT6	Blue data signal (MSB)	
32	VSHD	-	Digital power supply	See Section 2.1, “External Power Supplies” on page 8
33	DGND	V_{SS}	Digital ground	Ground pin of S1D13708
34	PS	GPIO0	Power save signal	
35	LP	FPLINE	Data latch signal of source driver	
36	DCLK	FPSHIFT	Data sampling clock signal	
37	LBR	-	Selection for horizontal scanning direction	Connect to VSHD (left / right scanning)
38	SPR	-	Sampling start signal for right / left scanning	Right to left scanning not supported
39	VSHA	-	Analog power supply	See Section 2.1, “External Power Supplies” on page 8
40	V0	-	Standard gray scale voltage (black)	See Section 2.1, “External Power Supplies” on page 8
41	V1	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
42	V2	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
43	V3	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
44	V4	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
45	V5	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
46	V6	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
47	V7	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
48	V8	-	Standard gray scale voltage	See Section 2.1, “External Power Supplies” on page 8
49	V9	-	Standard gray scale voltage (white)	See Section 2.1, “External Power Supplies” on page 8
50	AGND	V_{SS}	Analog ground	Ground pin of S1D13708

3 Connecting to the Sharp LQ031B1DDxx HR-TFT

3.1 External Power Supplies

The S1D13708 provides all necessary data and control signals to connect to the Sharp LQ031B1DDxx 160x160 HR-TFT panel(s). However, it does not provide any of the voltages required for the backlight, gray scaling, gate driving, or for the digital and analog supplies. Therefore, external supplies must be designed for any device utilizing the LQ031B1DDxx.

The LQ031B1DDxx (160x160) has the same voltage requirements as the LQ039Q2DS01 (320x240). All the circuits used to generate the various voltages for the LQ039Q2DS01 panel also apply to the LQ031B1DDxx panel. This section provides additional circuits for generating some of these voltages.

3.1.1 Gray Scale Voltages for Gamma Correction

The standard gray scale voltages can be generated using a precise resistor divider network as described in Section 2.1.1, “Gray Scale Voltages for Gamma Correction” on page 8. Alternately, they can be generated using a Sharp gray scale IC. The Sharp IR3E203 eliminates the large resistor network used to provide the 10 gray scale voltages and combines their function into a single IC.

The S1D13708 output signal REV is used to alternate the gray scale voltages and connects to the SW input of the IR3E203 IC. The COM signal is used in generating the gate driver panel AC voltage, V_{COM} and is explained in Section 3.1.4, “AC Gate Driver Power Supplies” on page 15. Figure 3-1: “Sharp LQ031B1DDxx Gray Scale Voltage (V0-V9) Generation” shows the circuit that generates the gray scale voltages using the Sharp IR3E203 IC.

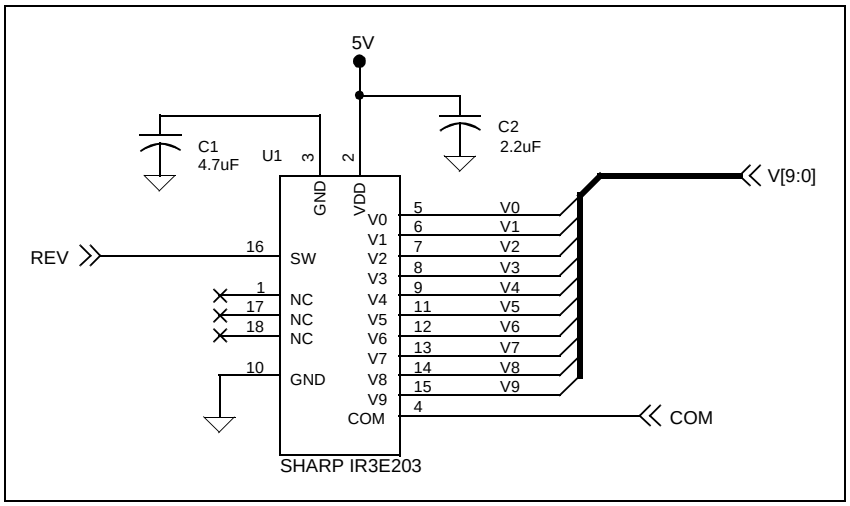


Figure 3-1: Sharp LQ031B1DDxx Gray Scale Voltage (V0-V9) Generation

3.1.2 Digital/Analog Power Supplies

The digital power supply (VSHD) must be connected to a 3.3V supply. The analog power supply (VSHA) must be connected to a 5.0V supply.

3.1.3 DC Gate Driver Power Supplies

See Section 2.1.3, “DC Gate Driver Power Supplies” on page 9 and Figure 2-2: “Panel Gate Driver DC Power Supplies,” on page 9 for details on generating V_{SS}, V_{DD}, and V_{CC}.

3.1.4 AC Gate Driver Power Supplies

See Section 2.1.4, “AC Gate Driver Power Supplies” on page 10 and Figure 2-3: “Panel Gate Driver AC Power Supplies,” on page 10 for details on generating V_{EE} and V_{COM}. If the Sharp IR3E203 is used to generate the gray scale voltages, the COM signal can be connected to the input of the F2C02E MOSFET instead of the buffered REV signal.

3.2 HR-TFT MOD Signal

See Section 2.2, “HR-TFT MOD Signal” on page 11 for details on controlling the MOD signal through software.

3.3 S1D13708 to LQ031B1DDxx Pin Mapping

Table 3-1: S1D13708 to LQ031B1DDxx Pin Mapping

LCD Pin No.	LCD Pin Name	S1D13708 Pin Name	Description	Remarks
1	VDD	-	Power supply of gate driver (high level)	See Section 3.1, “External Power Supplies” on page 14
2	VCC	-	Power supply of gate driver (logic high)	See Section 3.1, “External Power Supplies” on page 14
3	MOD	-	Control signal of gate driver	See Section 3.2, “HR-TFT MOD Signal” on page 15
4	MOD	-	Control signal of gate driver	See Section 3.2, “HR-TFT MOD Signal” on page 15
5	U/L	-	Selection for vertical scanning direction	Connect to VSHD (top / bottom scanning)
6	SPS	FPFRAME	Start signal of gate driver	
7	CLS	GPIO1	Clock signal of gate driver	
8	VSS	-	Power supply of gate driver (logic low)	See Section 3.1, “External Power Supplies” on page 14
9	VEE	-	Power supply of gate driver (low level)	See Section 3.1, “External Power Supplies” on page 14
10	VEE	-	Power supply of gate driver (low level)	See Section 3.1, “External Power Supplies” on page 14
11	VCOM	-	Common electrode driving signal	See Section 3.1, “External Power Supplies” on page 14
12	VCOM	-	Common electrode driving signal	See Section 3.1, “External Power Supplies” on page 14
13	SPL	GPIO3	Sampling start signal for left / right scanning	
14	R0	FPDAT11	Red data signal (LSB)	
15	R1	FPDAT10	Red data signal	
16	R2	FPDAT9	Red data signal	
17	R3	FPDAT2	Red data signal	
18	R4	FPDAT1	Red data signal	
19	R5	FPDAT0	Red data signal (MSB)	
20	G0	FPDAT14	Green data signal (LSB)	
21	G1	FPDAT13	Green data signal	
22	G2	FPDAT12	Green data signal	
23	G3	FPDAT5	Green data signal	
24	G4	FPDAT4	Green data signal	
25	G5	FPDAT3	Green data signal (MSB)	

Table 3-1: S1D13708 to LQ031B1DDxx Pin Mapping (Continued)

LCD Pin No.	LCD Pin Name	S1D13708 Pin Name	Description	Remarks
26	B0	FPDAT17	Blue data signal (LSB)	
27	B1	FPDAT16	Blue data signal	
28	B2	FPDAT15	Blue data signal	
29	B3	FPDAT8	Blue data signal	
30	B4	FPDAT7	Blue data signal	
31	B5	FPDAT6	Blue data signal (MSB)	
32	VSHD	-	Digital power supply	See Section 3.1, “External Power Supplies” on page 14
33	DGND	V_{SS}	Digital ground	Ground pin of S1D13708
34	PS	GPIO0	Power save signal	
35	LP	FPLINE	Data latch signal of source driver	
36	DCLK	FPSHIFT	Data sampling clock signal	
37	LBR	-	Selection for horizontal scanning direction	Connect to VSHD (left / right scanning)
38	SPR	-	Sampling start signal for right / left scanning	Right to left scanning not supported
39	VSHA	-	Analog power supply	See Section 3.1, “External Power Supplies” on page 14
40	V0	-	Standard gray scale voltage (black)	See Section 3.1, “External Power Supplies” on page 14
41	V1	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
42	V2	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
43	V3	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
44	V4	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
45	V5	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
46	V6	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
47	V7	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
48	V8	-	Standard gray scale voltage	See Section 3.1, “External Power Supplies” on page 14
49	V9	-	Standard gray scale voltage (white)	See Section 3.1, “External Power Supplies” on page 14
50	AGND	V_{SS}	Analog ground	Ground pin of S1D13708

4 Test Software

Test utilities and display drivers are available for the S1D13708. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13708CFG, or by directly modifying the source. The display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13708 test utilities and display drivers are available from your sales support contact or www.erd.epson.com.

5 References

5.1 Documents

- Sharp Electronics Corporation, *LQ039Q2DS01 Specification.*
- Sharp Electronics Corporation, *LQ031B1DDxx Specification.*
- Epson Research and Development, Inc., *S1D13708 Hardware Functional Specification*, Document Number X39A-A-001-xx.
- Epson Research and Development, Inc., *S1D13708 Programming Notes and Examples*, Document Number X39A-G-003-xx.

5.2 Document Sources

- Sharp Electronics Corporation Website: http://www.sharpsma.com.
- Epson Research and Development Website: http://www.erd.epson.com/.

6 Technical Support

6.1 EPSON LCD Controllers (S1D13708)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com/

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

6.2 Sharp HR-TFT Panel

http://www.sharpsma.com

EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

Interfacing to the Motorola RedCap2

Document Number: X39A-G-014-01

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to provide an interface between the S1D13708 Embedded Memory LCD Controller and the Motorola REDCAP2 processor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Research and Development Website at www.erd.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Interfacing to the REDCAP2

2.1 The REDCAP2 System Bus

REDCAP2 integrates a RISC microprocessor (MCU) and a general purpose digital signal processor (DSP) on a single chip. The External Interface Module (EIM) handles the interface to external devices. This section provides an overview of the operation of the REDCAP2 bus in order to establish interface requirements.

2.2 Overview

REDCAP2 uses a 22-bit address bus (A[21:0]) and 16-bit data bus (D[15:0]). All IO is synchronous to a square wave reference clock called CKO. The CKO source can be the DSP clock or the MCU clock and is selected/disabled in the Clock Control Register (CKCTL).

REDCAP2 can generate up to 6 independent chip select outputs. Each chip select has a memory range of 16M bytes and can be independently programmed for wait-states and port size.

Note
REDCAP2 does not provide a wait or termination acknowledge signal to external devices. Therefore, all external devices must guarantee a fixed cycle length.

2.3 Bus Transactions

The chip initiates a data transfer by placing the memory address on address lines A0 through A21. Several control signals are provided with the memory address.

- R/$\overline{W}$ — set high for read cycles and low for write cycles.
- $\overline{EB0}$ — active low signal indicates access to data byte 0 (D[15:8]) during read or write cycles.
- $\overline{EB1}$ — active low signal indicates access to data byte 1 (D[7:0]) during read or write cycles.
- $\overline{OE}$ — active low signal indicates read accesses and enables slave devices to drive the data bus.

Figure 2-1: “REDCAP2 Memory Read Cycle” on page 9 illustrates a typical memory read cycle on the REDCAP2 bus.

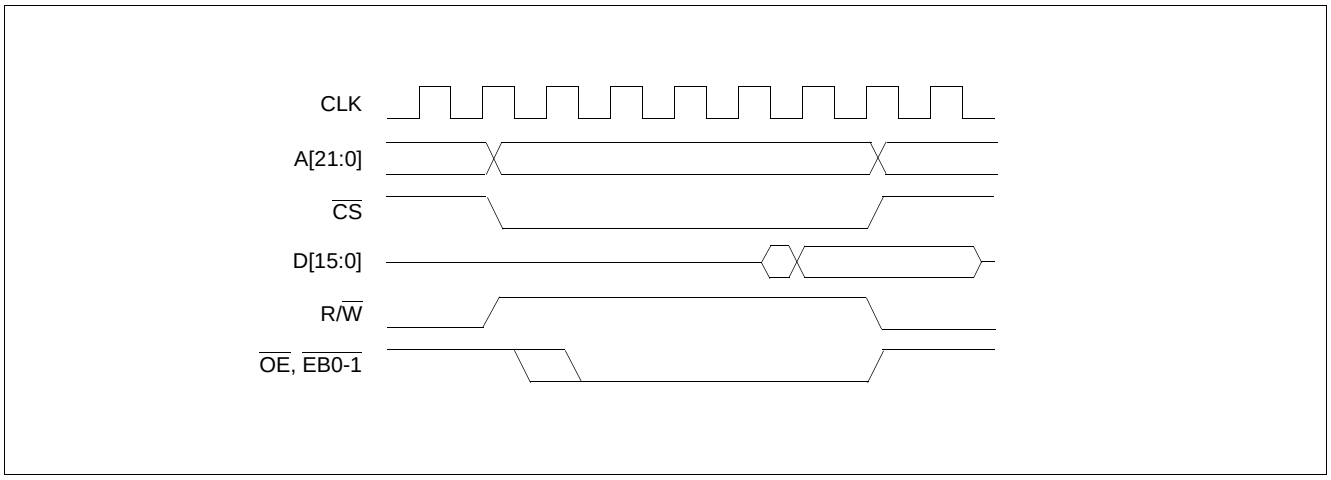


Figure 2-1: REDCAP2 Memory Read Cycle

Figure 2-2: “REDCAP2 Memory Write Cycle” on page 9 illustrates a typical memory write cycle on the REDCAP2 bus.

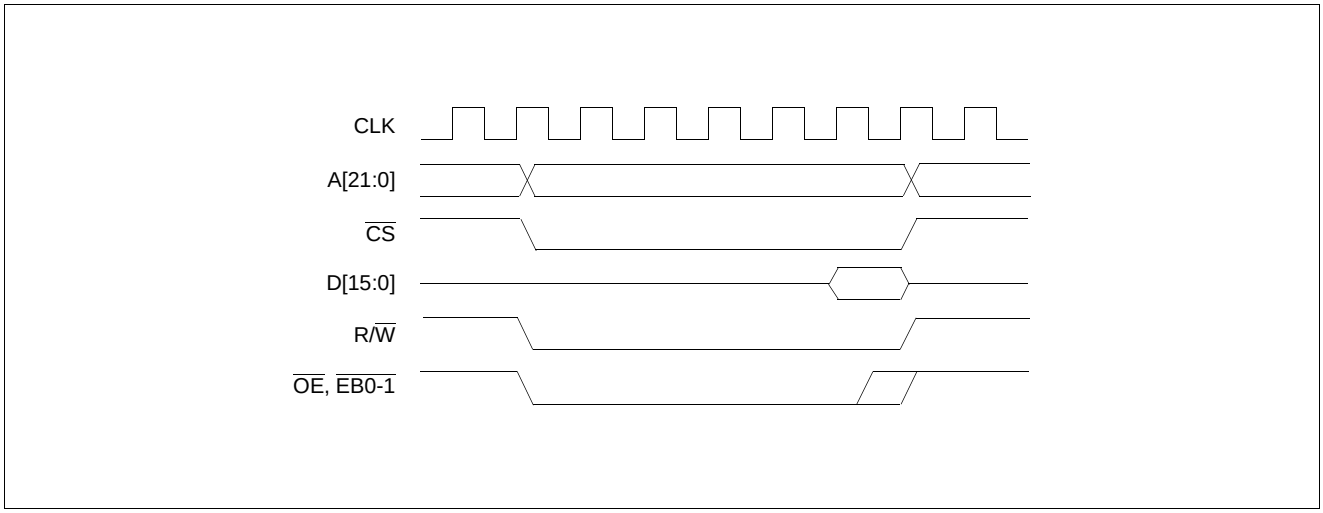


Figure 2-2: REDCAP2 Memory Write Cycle

3 S1D13708 Host Bus Interface

The S1D13708 implements a 16-bit native REDCAP2 host bus interface which is used to interface to the REDCAP2 processor.

The REDCAP2 host bus interface is selected by the S1D13708 on the rising edge of RESET#. After releasing reset, the bus interface signals assume their selected configuration. For details on S1D13708 configuration, see Section 4.3, “S1D13708 Hardware Configuration” on page 15.

3.1 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13708 Pin Names	REDCAP2
AB[16:0]	A[16:0]
DB[15:0]	D[15:0]
WE1#	$\overline{EB0}$
M/R#	A17
CS#	REDCAP2 Internal Chip Select
CLKI	CKO
BS#	Connected to IO V_{DD}
RD/WR#	R/$\overline{W}$
RD#	$\overline{OE}$
WE0#	$\overline{EB1}$
WAIT#	N/A
RESET#	$\overline{RST\_OUT}$

3.2 Host Bus Interface Signals

The Host Bus Interface requires the following signals:

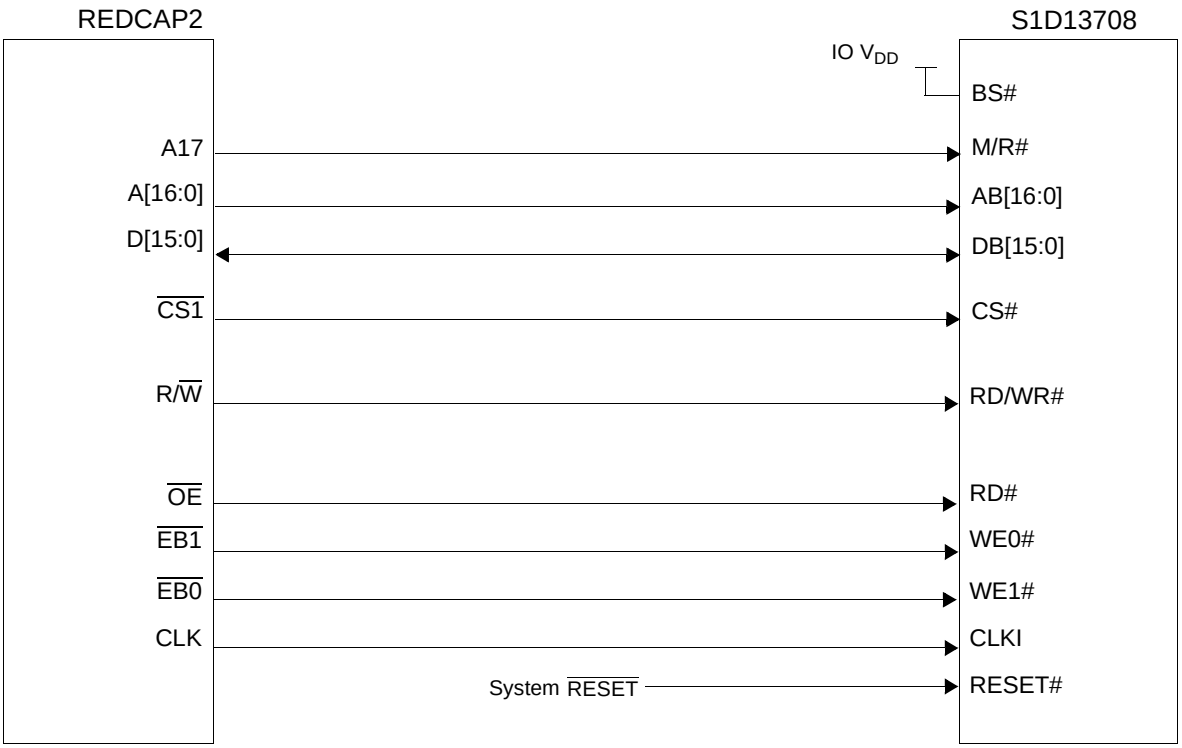
- CLKI is a clock input which is required by the S1D13708 host bus interface and connects to CKO of the REDCAP2.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the REDCAP2 bus address (A[16:0]) and data bus (D[15:0]), respectively. CNF[2:0] and CNF4 must be set to select the REDCAP2 host bus interface with big endian mode.
- M/R# (memory/register) selects between memory or register access. It may be connected to an address line, allowing REDCAP2 bus address A17 to be connected to the M/R# line.
- CS# (Chip Select) must be driven low whenever the S1D13708 is accessed by the REDCAP2 bus.
- RD/WR# connects to R/$\overline{W}$ which indicates whether a read or a write access is being performed on the S1D13708.
- WE1# and WE0# connect to $\overline{EB0}$ and $\overline{EB1}$ (Enable Byte 0 and 1) for byte steering.
- RD# connects to $\overline{OE}$ (Output Enable). This signal must be driven by the REDCAP2 bus to indicate the bus access is a read and enables slave devices to drive the data bus with read data.
- The BS# and WAIT# signals are not needed for this bus interface, they should be connected to IO V_{DD}.

4 REDCAP2 to S1D13708 Interface

4.1 Hardware Description

The interface between the S1D13708 and the REDCAP2 requires no external glue logic. The information in this section describes the environment necessary to connect the S5U13708B00C Evaluation Board and the Motorola DSP56654 Application Development Module (ADM). For a list of connections for the pins and signals of the REDCAP2 see Table 4-1:,"List of Connections for REDCAP2 ADM" on page 13.

The following figure demonstrates a typical implementation of the S1D13708 to the REDCAP2 interface.



*Note: This example uses $\overline{CS1}$. $\overline{CSn}$ can be any of $\overline{CS0}$-$\overline{CS4}$.

**Note: When connecting the S1D13708 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13708 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 4-1: Typical Implementation of REDCAP2 to S1D13708 Interface

4.2 Hardware Connections

The following table details the connections for the pins and signals of the REDCAP2.

Table 4-1: List of Connections for REDCAP2 ADM

REDCAP2 Signal Name	REDCAP2ADS Connector and Pin Name	S1D13708 Signal Name
A17	P9-34	M/R#
A16	P9-33	AB20
A15	P9-32	AB19
A14	P9-31	AB18
A13	P9-30	AB17
A12	P9-29	AB16
A11	P9-28	AB15
A10	P9-27	AB14
A9	P9-26	AB13
A8	P9-25	AB12
A7	P9-24	AB11
A6	P9-23	AB10
A5	P9-22	AB9
A4	P9-21	AB8
A3	P9-20	AB7
A2	P9-19	AB6
A1	P9-18	AB5
A0	P9-17	AB4
D15	P9-16	DB15
D14	P9-15	DB14
D13	P9-14	DB13
D12	P9-13	DB12
D11	P9-12	DB11
D10	P9-11	DB10
D9	P9-10	DB9
D8	P9-9	DB8
D7	P9-8	DB7
D6	P9-7	DB6
D5	P9-6	DB5
D4	P9-5	DB4
D3	P9-4	DB3
D2	P9-3	DB2
D1	P9-2	DB1
D0	P9-1	DB0
RES_OUT	P24-6	RESET#
CLK0	P24-3	BUSCLK

Table 4-1: List of Connections for REDCAP2 ADM (Continued)

REDCAP2 Signal Name	REDCAP2ADS Connector and Pin Name	S1D13708 Signal Name
$\overline{CS1}$	P9-40	CS#
R/$\overline{W}$	P9-47	RD/WR#
$\overline{OE}$	P9-48	RD#
$\overline{EB1}$	P9-46	WE0#
$\overline{EB0}$	P9-45	WE1#
Gnd	P24-20 / P9-50	Vss

Note

Pin 5 and pin 13 of U28 on the ADM must be connected to $V_{DD.}$ This ensures that the DIR signal of transceivers U17 and U18 is low only during read access, even when EBC in the CS1 Control Register is set to 0 when connecting to the S1D13708.

4.3 S1D13708 Hardware Configuration

The S1D13708 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13708 to Motorola REDCAP2 microprocessor.

Table 4-2: Summary of Power-On/Reset Options

S1D13708 Pin Name	**state of this pin at rising edge of RESET# is used to configure:**(1/0)	
	1	**0**
CNF[4, 2:0]	1101 = REDCAP2 Big Endian Host Bus Interface	
CNF3	GPIO pins as inputs at power-on	GPIO pins as HR-TFT/ D-TFD outputs
CNF5	WAIT# is active high	WAIT# is active low
CNF[7:6]	CLKI to BCLK divide select: CNF7 CNF6 CLKI to BCLK Divide Ratio 0 0 1 : 1 0 1 2 : 1 1 0 3 : 1 1 1 4 : 1	

= configuration for REDCAP2 microprocessor

4.4 Register/Memory Mapping

The S1D13708 is a memory mapped device. The S1D13708 uses two 128K byte blocks which are selected using A17 from the REDCAP2 bus (A17 is connected to the S1D13708 M/R# pin). The internal registers occupy the first 128K byte block and the 80K byte display buffer occupies the second 128K byte block. In this example, the S1D13708 internal registers are accessed starting at address 4100 0000h and the display buffer is accessed starting at address 4102 0000h.

Each Chip Select on the REDCAP2 is allocated a 16M byte block. However, the S1D13708 only needs a 256K byte block of memory to accommodate its register set and 80K byte display buffer. For this reason, only address bits A[17:0] are used while A[21:18] are ignored. The S1D13708's memory and register are aliased every 256K bytes in the 16M byte CS1 address space.

Note
If aliasing is not desirable, the upper addresses must be fully decoded.

4.5 REDCAP2 Chip Select Configuration

In this example, Chip Select 1 controls the S1D13708. The following options are selected in the CS1 Control Register.

- CSEN = 1 — Chip Select function enabled.
- WP = 0 — writes allowed.
- SP = 0 — user mode access allowed.
- DSZ = 10 — 16-bit Port.
- EBC = 0 — assert $\overline{\text{EB0-1}}$ for both reads and writes.
- WEN = 1 — $\overline{\text{EB0-1}}$ negated half a clock earlier during write cycle.
- OEA = 1 — $\overline{\text{OE}}$ asserted half a clock later during a read cycle.
- CSA = 0 — Chip Select asserted as early as possible. No idle cycle inserted between back-to-back external transfers.
- EDC = 1 — an idle cycle is inserted after a read cycle for back-to-back external transfers, unless the next cycle is a read cycle to the same $\overline{\text{CS}}$ bank.
- WWS = 0 — same length for reads and writes.

5 Software

Test utilities and display drivers are available for the S1D13708. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13708CFG, or by directly modifying the source. The display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13708 test utilities and display drivers are available from your sales support contact or www.erd.epson.com.

6 References

6.1 Documents

- Motorola Inc., *REDCAP2 Digital Signal Processor Integrated With MCU Product Specifications Rev. 1.2ext.*
- Epson Research and Development, Inc., *S1D13708 Hardware Functional Specification*, Document Number X39A-A-001-xx.
- Epson Research and Development, Inc., *S5U13708B00C Rev. 1.0 Evaluation Board User Manual*, Document Number X39A-G-004-xx.
- Epson Research and Development, Inc., *S1D13708 Programming Notes and Examples*, Document Number X39A-G-003-xx.

6.2 Document Sources

- Motorola Inc. Literature Distribution Center: (800) 441-2447.
- Motorola Inc. Website: http://www.motorola.com/.
- Epson Research and Development Website: http://www.erd.epson.com/.

7 Technical Support

7.1 EPSON LCD/CRT Controllers (S1D13708)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com/

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

7.2 Motorola REDCAP2 Processor

- Motorola Design Line, (800) 521-6274.
- Local Motorola sales office or authorized distributor.

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

Interfacing to 8-bit Processors

Document Number: X39A-G-015-01

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13708 Embedded Memory LCD Controller and 8-bit processors. This document is not intended to cover all possible implementation, but provides a generic example of how such an interface can be accomplished.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Research and Development website at http://www.erd.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Interfacing to an 8-bit Processor

2.1 The Generic 8-bit Processor System Bus

Although the S1D13708 does not directly support an 8-bit CPU, an 8-bit interface can be achieved with minimal external logic.

Typically, the bus of an 8-bit microprocessor is straight forward with minimal CPU and system control signals. To connect a memory mapped device such as the S1D13708, only the write, read, and wait control signals, plus the data and address lines, need to be interfaced. Since the S1D13708 is a 16-bit device, some external logic is required.

3 S1D13708 Host Bus Interface

The S1D13708 directly supports multiple processors. The S1D13708 implements a 16-bit Generic #2 Host Bus Interface which can be adapted for use with an 8-bit processor.

The Generic #2 Host Bus Interface is selected by the S1D13708 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13708 configuration, see Section 4.2, "S1D13708 Hardware Configuration" on page 12.

The S1D13708 clock (CLKI) is taken from the system host bus. The system clock source will drive all required internal clocks. If they are not used, the CLKI2 and XTAL inputs should be tied to ground.

3.1 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13708 Pin Names	Generic #2	Comments
AB[16:0]	A[16:0]	—
DB[15:0]	D[15:0]	—
WE1#	Byte High Enable (BHE#)	External decode required
CS#	Chip Select	External decode required
M/R#	Memory/Register Select	External decode required
CLKI	BUSCLK	—
BS#	connect to IO V_{DD}	—
RD/WR#	connect to IO V_{DD}	—
RD#	RD#	—
WE0#	WE#	—
WAIT#	WAIT#	—
RESET#	Inverted RESET	—

3.2 Host Bus Interface Signals

The S1D13708 Generic #2 Host Bus Interface requires the following signals from an 8-bit processor.

- CLKI is a clock input which is required by the S1D13708 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock.
- The address inputs AB[16:0] connect directly to the 8-bit processor address lines (A[16:0). If the specific 8-bit processor cannot implement all 17 address lines required by the S1D13708, only a portion of the 80K byte S1D13708 display buffer is accessible. For example, if only AB[15:0] are supported, only the first 64K byte of the display buffer is available.
- The data bus DB[15:0] must be connected so that the 8-bit processor data lines (D[7:0]) are connected to both DB[15:8] and DB[7:0] of the S1D13708. CNF4 must be set to select little endian mode.
- Chip Select (CS#) is driven by decoding the high-order address lines to select the proper register and memory address space.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address A17 to be connected to the M/R# line.

  Note
  If A17 is unavailable on the 8-bit processor, an external decode must be used to generate the M/R# signal.

- BHE# is the high byte enable for both read and write cycles and connects to the high byte chip select signal.

  Note
  In an 8-bit environment, this signal is driven by inverting address line A0 thus indicating that odd addresses are to be read/write on the high byte of the data bus.

- WE# connects to WE# (the write enable signal) and must be driven low when the 8-bit processor is writing data to the S1D13708.
- RD# connects to RD# (the read enable signal) and must be driven low when the 8-bit processor is reading data from the S1D13708.
- WAIT# is a signal output from the S1D13708 that indicates the 8-bit processor must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since host CPU bus accesses to the S1D13708 may occur asynchronously to the display update, it is possible that contention may occur in accessing the 13708 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete. This signal is active low and may need to be inverted if the host CPU wait state signal is active high.
- The Bus Status (BS#) and Read/Write (RD/WR#) signals are not used in this implementation of a generic 8-bit processor using the Generic #2 Host Bus Interface. These pins must be tied high (connected to IO V_{DD}).

4 8-Bit Processor to S1D13708 Interface

4.1 Hardware Connections

The interface between the S1D13708 and an 8-bit processor requires minimal glue logic. A decoder is used to generate the chip select for the S1D13708 based on where the S1D13708 is mapped into memory. Alternatively, if the processor supports a chip select module, it can be programmed to generate a chip select for the S1D13708 without the need of an address decoder.

An inverter inverts A0 to generate the BHE# signal for the S1D13708. BS# (bus start) and RD/WR# are not used by the Generic #2 Host Bus Interface and must be tied high (connected to IO V_{DD}).

In order to support an 8-bit processor with a 16-bit peripheral, the low and high order bytes of the data bus must be connected together. The following diagram shows a typical implementation of an 8-bit processor to S1D13708 interface.

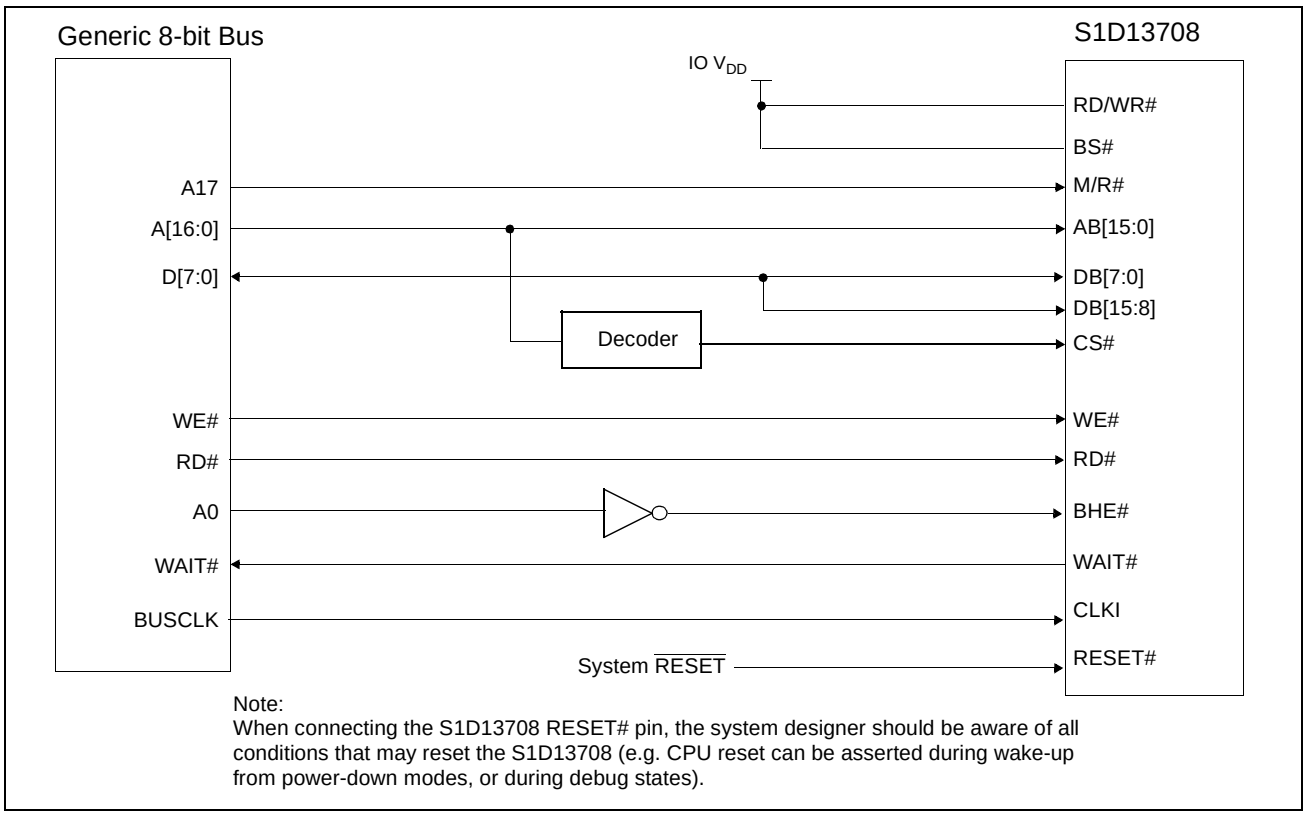


Figure 4-1: Typical Implementation of 8-bit Processor to S1D13708 Interface

4.2 S1D13708 Hardware Configuration

The S1D13708 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13708 to generic 8-bit processor.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13708 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[4, 2:0]	0100 = Generic #2 Little Endian Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table 4-2: "CLKI to BCLK Divide Selection" for recommended setting	

= configuration for generic 8-bit processor

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for generic 8-bit processor

4.3 Register/Memory Mapping

The S1D13708 is a memory mapped device. The S1D13708 uses two 128K byte blocks which are selected using A17 from the 8-bit processor (A17 is connected to the S1D13708 M/R# pin). The internal registers occupy the first 128K byte block and the 80K byte display buffer occupies the second 128K byte block.

An external decoder can be used to decode the address lines and generate a chip select for the S1D13708 whenever the selected 128k byte memory block is accessed. If the processor supports a general chip select module, its internal registers can be programmed to generate a chip select for the S1D13708 whenever the S1D13708 memory block is accessed.

5 Software

Test utilities and display drivers are available for the S1D13708. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13708CFG, or by directly modifying the source. The display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13708 test utilities and display drivers are available from your sales support contact or www.erd.epson.com.

6 References

6.1 Documents

- Epson Research and Development, Inc., *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.
- Epson Research and Development, Inc., *S5U13708B00C Rev. 1.0 Evaluation Board User Manual*, document number X39A-G-004-xx.
- Epson Research and Development, Inc., *S1D13708 Programming Notes and Examples*, Document Number X39A-G-003-xx.

6.2 Document Sources

- Epson Research and Development Website: http://www.erd.epson.com/.

7 Technical Support

7.1 EPSON LCD Controllers (S1D13708)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com/

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

THIS PAGE LEFT BLANK

EPSON®

S1D13708 Embedded Memory LCD Controller

Interfacing to the Motorola MC68VZ328 Dragonball Microprocessor

Document Number: X39A-G-016-01

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to interface the S1D13708 Embedded Memory LCD Controller and the Motorola MC68VZ328 Dragonball VZ microprocessor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Research and Development website at http://www.erd.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Interfacing to the MC68VZ328

2.1 The MC68VZ328 System Bus

The Motorola MC68VZ328 “Dragonball VZ” is the third generation in the Dragonball microprocessor family. The Dragonball VZ is an integrated controller designed for handheld products. It is based upon the FLX68000 microprocessor core and uses a 24-bit address bus and 16-bit data bus. The Dragonball VZ is faster than its predecessors and the DRAM controller now supports SDRAM. The bus interface consists of all the standard MC68000 bus interface signals except $\overline{AS}$, plus some new signals intended to simplify the interface to typical memory and peripheral devices. The 68000 signals are multiplexed with IrDA, SPI and LCD controller signals.

The MC68000 bus control signals are well documented in the Motorola user manuals, and are not be described here. The new signals are as follows.

- Output Enable ($\overline{OE}$) is asserted when a read cycle is in progress. It is intended to connect to the output enable control signal of a typical static RAM, EPROM, or Flash EPROM device.
- Upper Write Enable and Lower Write Enable ($\overline{UWE}$ / $\overline{LWE}$) are asserted during memory write cycles for the upper and lower bytes of the 16-bit data bus. They may be directly connected to the write enable inputs of a typical memory device.

2.2 Chip-Select Module

The MC68VZ328 can generate up to 8 chip select outputs which are organized into four groups (A through D).

Each chip select group has a common base address register and address mask register allowing the base address and block size of the entire group to be set. In addition, each chip select within a group has its own address compare and address mask register to activate the chip select for a subset of the group’s address block. Each chip select may also be individually programmed to control an 8 or 16-bit device. Lastly, each chip select can either generate from 0 through 6 wait states internally, or allow the memory or peripheral device to terminate the cycle externally using the standard MC68000 $\overline{DTACK}$ signal.

Chip select groups A and B are used to control ROM, SRAM, and Flash memory devices and have a block size of 128K bytes to 16M bytes. Chip select A0 is active immediately after reset and is a global chip select so it is typically used to control a boot EPROM device. A0 ceases to decode globally once its chip select registers are programmed. Groups C and D are special in that they can also control DRAM interfaces. These last two groups have block size of 32K bytes to 4M bytes.

3 S1D13708 Host Bus Interface

The S1D13708 directly supports multiple processors. The S1D13708 implements a Dragonball Host Bus Interface which directly supports the Motorola MC68VZ328 microprocessor.

The Dragonball Host Bus Interface is selected by the S1D13708 on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13708 configuration, see Section 4.2, "S1D13708 Hardware Configuration" on page 12.

The S1D13708 clock (CLKI) is taken from the system host bus. The system clock source will drive all required internal clocks. If they are not used, the CLKI2 and XTAL inputs should be tied to ground.

3.1 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13708 Pin Names	Motorola MC68VZ328
AB[16:0]	A[16:0]
DB[15:0]	D[15:0]
WE1#	$\overline{\text{UWE}}$
CS#	$\overline{\text{CSx}}$
M/R#	External Decode
CLKI	CLKO
BS#	Connect to IO_{VDD} from the S1D13708
RD/WR#	Connect to IO_{VDD} from the S1D13708
RD#	$\overline{\text{OE}}$
WE0#	$\overline{\text{LWE}}$
WAIT#	$\overline{\text{DTACK}}$
RESET#	System $\overline{\text{RESET}}$

3.2 Host Bus Interface Signals

The Host Bus Interface requires the following signals.

- CLKI is a clock input required by the S1D13708 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For this example, CLK0 from the Motorola MC68VZ328 is used for CLKI.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the MC68VZ328 address (A[16:0]) and data bus (D[15:0]), respectively. CNF4 must be set to one to select big endian mode.
- Chip Select (CS#) must be driven low by one of the Dragonball VZ chip select outputs from the chip select module whenever the S1D13708 is accessed by the MC68VZ328.
- M/R# (memory/register) selects between memory or register accesses. This signal is generated by the external address decode circuitry. For this example, M/R# may be connected to an address line, allowing system address A17 to be connected to the M/R# line.
- WE0# connects to $\overline{\text{LWE}}$ (the low data byte write strobe enable of the MC68VZ328) and is asserted when valid data is written to the low byte of a 16-bit device.
- WE1# connects to $\overline{\text{UWE}}$ (the upper data byte write strobe enable of the MC68VZ328) and is asserted when valid data is written to the high byte of a 16-bit device.
- RD# connects to $\overline{\text{OE}}$ (the read output enable of the MC68VZ328) and is asserted during a read cycle of the MC68VZ328 microprocessor.
- RD/WR# is not used for the Dragonball host bus interface and must be tied high to IO V_{DD}.
- WAIT# connects to $\overline{\text{DTACK}}$ and is a signal which is output from the S1D13708 indicating the MC68VZ328 must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. The MC68VZ328 accesses to the S1D13708 may occur asynchronously to the display update.
- BS# is not used for the Dragonball host bus interface and must be tied high to IO V_{DD}.

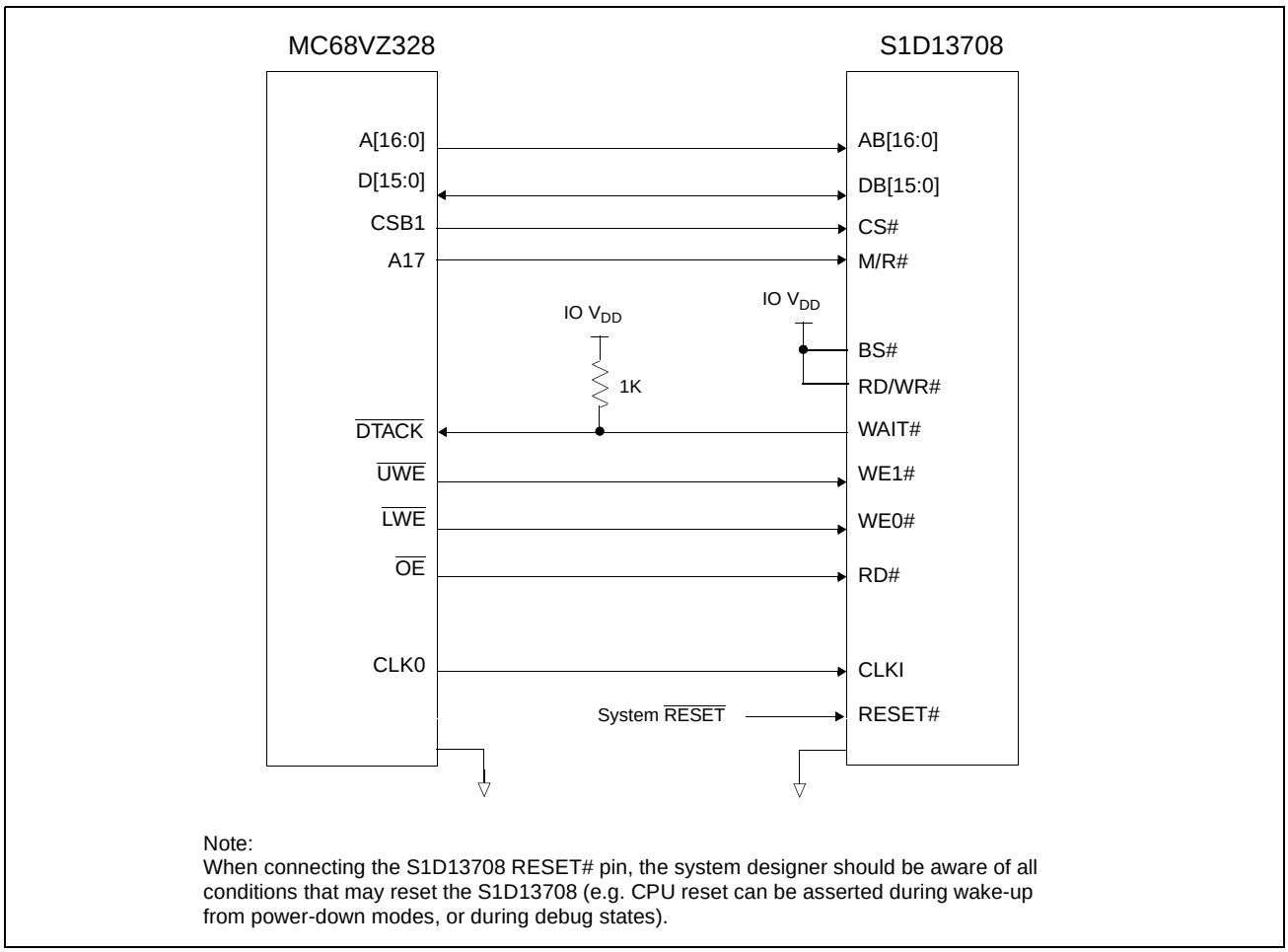
4 MC68VZ328 to S1D13708 Interface

4.1 Hardware Description

The interface between the S1D13708 and the MC68VZ328 does not requires any external glue logic. Chip select module B is used to provide the S1D13708 with a chip select and A17 is used to select between memory and register accesses.

In this example, the $\overline{\text{DTACK}}$ signal is made available for the S1D13708. Alternately, the S1D13708 can guarantee a maximum cycle length that the Dragonball VZ handles by inserting software wait states (see Section 4.2.2, "MC68VZ328 Chip Select and Pin Configuration" on page 13). A single resistor is used to speed up the rise time of the WAIT# ($\overline{\text{DTACK}}$) signal when terminating the bus cycle.

The following diagram shows a typical implementation of the MC68VZ328 to S1D13708 using the Dragonball host bus interface. For further information on the Dragonball Host Bus interface and AC Timing, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.



Note:
When connecting the S1D13708 RESET# pin, the system designer should be aware of all conditions that may reset the S1D13708 (e.g. CPU reset can be asserted during wake-up from power-down modes, or during debug states).

Figure 4-1: Typical Implementation of MC68VZ328 to S1D13708 Interface

4.2 S1D13708 Hardware Configuration

The S1D13708 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13708 to Motorola MC68VZ328 microprocessor.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13708 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[4, 2:0]	1110 = Dragonball Big Endian Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table 4-2: "CLKI to BCLK Divide Selection" for recommended settings	

= configuration for MC68VZ328 microprocessor

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	reserved

= recommended setting for MC68VZ328 microprocessor

4.2.1 Register/Memory Mapping

The S1D13708 requires two 128K byte segments in memory for the display buffer and its internal registers. To accommodate this block size, it is preferable (but not required) to use one of the chip selects from groups A or B. Groups A and B can have a size range of 128K bytes to 16M bytes and groups C and D have a size range of 32K bytes to 16M bytes. Therefore, any chip select other than CSA0 would be suitable for the S1D13708 interface.

In the example interface, chip select CSB1 controls the S1D13708. A 256K byte address space is used with the S1D13708 internal registers occupying the first 128K byte block and the 80K byte display buffer located in the second 128K byte block. A17 from the MC68VZ328 is used to select between these two 128K byte blocks.

4.2.2 MC68VZ328 Chip Select and Pin Configuration

The chip select used to map the S1D13708 (in this example CSB1) must have its RO (Read Only) bit set to 0, its BSW (Bus Data Width) set to 1 for a 16-bit bus, and the WS (Wait states) bits should be set to 111b to allow the S1D13708 to terminate bus cycles externally with $\overline{\text{DTACK}}$. The $\overline{\text{DTACK}}$ pin function must be enabled with Register FFFFF433, Port G Select Register, bit 0.

If $\overline{\text{DTACK}}$ is not used, then the WS bits should be set to either 4, 6, 10, or 12 software wait states depending on the divide ratio between the S1D13708 MCLK and BCLK. The WS bits should be set as follows.

Table 4-3: WS Bit Programming

S1D13708 MCLK to BCLK Divide Ratio	WS Bits (wait states)
MCLK = BCLK	6
MCLK = BCLK ÷ 2	9
MCLK = BCLK ÷ 3	12

5 Software

Test utilities and display drivers are available for the S1D13708. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13708CFG, or by directly modifying the source. The display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13708 test utilities and display drivers are available from your sales support contact or www.erd.epson.com.

6 References

6.1 Documents

- Motorola Inc., *MC68VZ328 DragonBall-VZ® Integrated Processor User's Manual*, Motorola Publication no. MC683VZ28UM; available on the Internet at http://www.motorola.com/.
- Epson Research and Development, Inc., *S1D13708 Hardware Functional Specification*, Document Number X39A-A-001-xx.
- Epson Research and Development, Inc., *S5U13708B00C Rev. 1.0 Evaluation Board User Manual*, Document Number X39A-G-004-xx.
- Epson Research and Development, Inc., *Programming Notes and Examples*, Document Number X39A-G-003-xx.

6.2 Document Sources

- Motorola Inc. Literature Distribution Center: (800) 441-2447.
- Motorola Inc. Website: http://www.motorola.com/.
- Epson Research and Development Website: http://www.erd.epson.com/.

7 Technical Support

7.1 EPSON LCD/CRT Controllers (S1D13708)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com/

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

7.2 Motorola MC68VZ328 Processor

- Motorola Design Line, (800) 521-6274.
- Local Motorola sales office or authorized distributor.

EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

Interfacing to the Intel StrongARM SA-1110 Microprocessor

Document Number: X39A-G-019-01

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software environment required to provide an interface between the S1D13708 Embedded Memory LCD Controller and the Intel StrongARM SA-1110 Microprocessor.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Research and Development website at www.erd.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Interfacing to the StrongARM SA-1110 Bus

2.1 The StrongARM SA-1110 System Bus

The StrongARM SA-1110 microprocessor is a highly integrated communications microcontroller that incorporates a 32-bit StrongARM RISC processor core. The SA-1110 is ideally suited to interface to the S1D13708 LCD controller and provides a high performance, power efficient solution for embedded systems.

2.1.1 StrongARM SA-1110 Overview

The SA-1110 system bus can access both variable-latency IO and memory devices. The SA-1110 uses a 26-bit address bus and a 32-bit data bus which can be used to access 16-bit devices. A chip select module with six chip select signals (each accessing 64M bytes of memory) allows selection of external devices. Only chip selects 3 through 5 (nCS[5:3]) may be used to select variable-latency devices which use RDY to extend access cycles. These chip selects are individually programmed in the SA-1110 memory configuration registers and can be configured for either a 16 or 32-bit data bus.

Byte steering is implemented using the four signals nCAS[3:0]. Each signal selects a byte on the 32-bit data bus. For example, nCAS0 selects bits D[7:0] and nCAS3 selects bits D[31:24]. For a 16-bit data bus, only nCAS[1:0] are used with nCAS0 selecting the low byte and nCAS1 selecting the high byte. The SA-1110 can be configured to support little or big endian mode.

2.1.2 Variable-Latency IO Access Overview

A data transfer is initiated when a memory address is placed on the SA-1110 system bus **and** a chip select signal (nCS[5:3]) is driven low. If all byte enable signals (nCAS[3:0]) are driven low, then a 32-bit transfer takes place. If only nCAS[1:0] are driven low, then a word transfer takes place through a 16-bit bus interface. If only one byte enable is driven low, then a byte transfer takes place on the respective data lines.

During a read cycle, the output enable signal (nOE) is driven low. A write cycle is specified by driving nOE high and driving the write enable signal (nWE) low. The cycle can be lengthened by driving RDY high for the time needed to complete the cycle.

2.1.3 Variable-Latency IO Access Cycles

The first nOE assertion occurs two memory cycles after the assertion of chip select (nCS3, nCS4, or nCS5). Two memory cycles prior to the end of minimum nOE or nWE assertion (RDF+1 memory cycles), the SA-1110 starts sampling the data ready input (RDY). Samples are taken every half memory cycle until **three consecutive samples** (at the rising edge, falling edge, and following rising edge of the memory clock) indicate that the IO device is ready for data transfer. Read data is latched one-half memory cycle after the third successful sample (on falling edge). Then nOE or nWE is deasserted on the next rising edge and the address may change on the subsequent falling edge. Prior to a subsequent data cycle, nOE or nWE remains deasserted for RDN+1 memory cycles. The chip select and byte selects (nCAS[1:0] for 16-bit data transfers), remain asserted for one memory cycle after the final nOE or nWE deassertion of the burst.

The SA-1110 is capable of burst cycles during which the chip select remains low while the read or write command is asserted, precharged and reasserted repeatedly.

Figure 2-1: illustrates a typical variable-latency IO access read cycle on the SA-1110 bus.

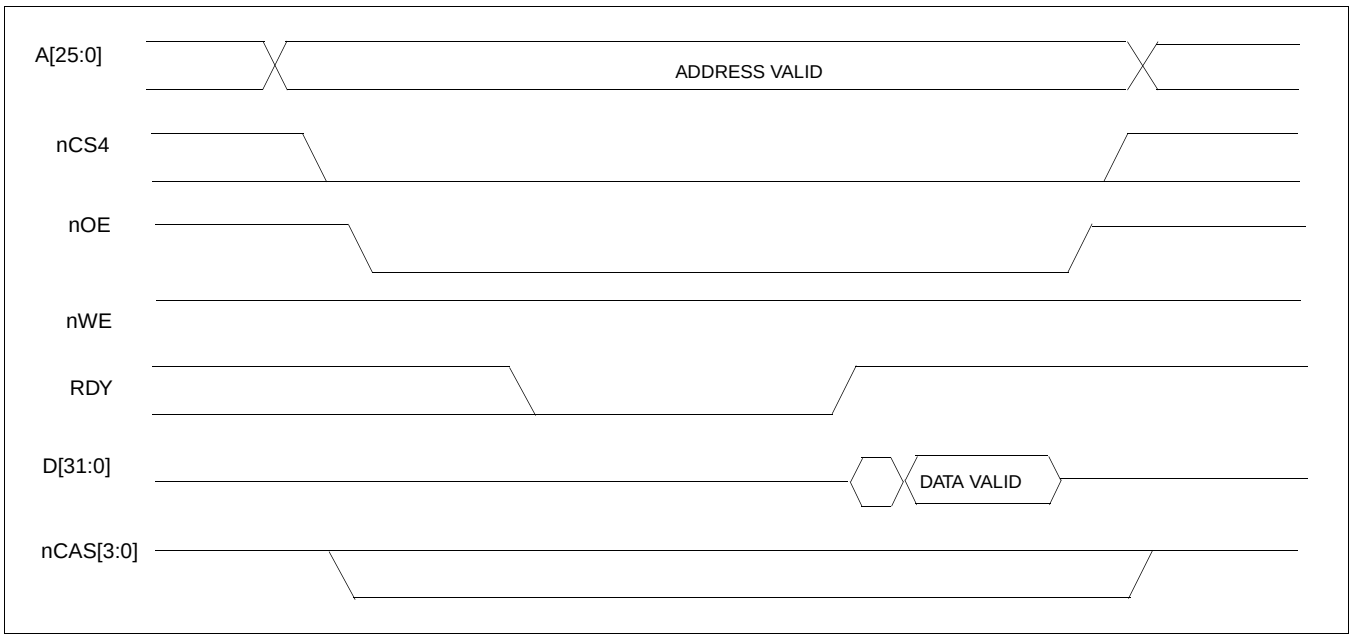


Figure 2-1: SA-1110 Variable-Latency IO Read Cycle

Figure 2-2: illustrates a typical variable-latency IO access write cycle on the SA-1110 bus.

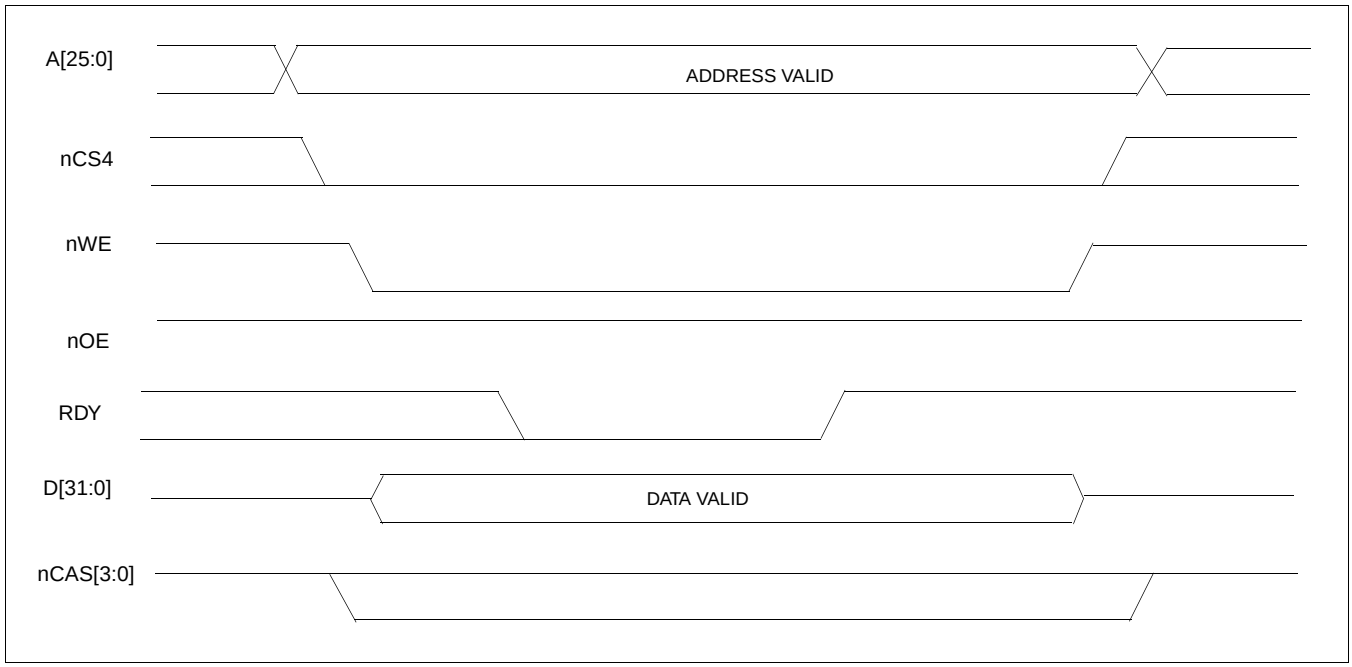


Figure 2-2: SA-1110 Variable-Latency IO Write Cycle

3 S1D13708 Host Bus Interface

The S1D13708 directly supports multiple processors. The S1D13708 implements a 16-bit Generic #2 Host Bus Interface which is most suitable for direct connection to the SA-1110.

The Generic #2 Host Bus Interface is selected by the S1D13708 on the rising edge of RESET#. After releasing reset the bus interface signals assume their selected configuration. For details on S1D13708 configuration, see Section 4.2, *"S1D13708 Hardware Configuration"* on page 14.

The S1D13708 clock (CLKI) is taken from the system host bus. The system clock source will drive all required internal clocks. If they are not used, the CLKI2 and XTAL inputs should be tied to ground.

3.1 Host Bus Interface Pin Mapping

The following table shows the functions of each Host Bus Interface signal.

Table 3-1: Host Bus Interface Pin Mapping

S1D13708 Pin Name	SA-1110
AB[16:0]	A[16:0]
DB[15:0]	D[15:0]
WE1#	nCAS1
M/R#	A17
CS#	nCS4
CLKI	SDCLK2
BS#	V_{DD}
RD/WR#	V_{DD}
RD#	nOE
WE0#	nWE
WAIT#	RDY
RESET#	system RESET

3.2 Host Bus Interface Signal Descriptions

The S1D13708 Generic #2 Host Bus Interface requires the following signals.

- CLKI is a clock input which is required by the S1D13708 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. For this example, it is driven by one of the SA-1110 signals SDCLK1 or SDCLK2 (The example implementation in this document uses SDCLK2). For further information, see Section 4.3, *"StrongARM SA-1110 Register Configuration"* on page 15.
- The address inputs AB[16:0], and the data bus DB[15:0], connect directly to the SA-1110 address bus (A[16:0]) and data bus (D[15:0]), respectively. CNF4 must be set to select little endian mode.
- M/R# (memory/register) selects between memory or register accesses. This signal may be connected to an address line, allowing system address A17 to be connected to the M/R# line.
- Chip Select (CS#) must be driven low by nCSx (where x is the SA-1110 chip select used) whenever the S1D13708 is accessed by the SA-1110.
- WE1# connects to nCAS1 (the high byte enable signal from the SA-1110) which in conjunction with address bit 0 allows byte steering of read and write operations.
- WE0# connects to nWE (the write enable signal from the SA-1110) and must be driven low when the SA-1110 is writing data to the S1D13708.
- RD# connects to nOE (the read enable signal from the SA-1110) and must be driven low when the SA-1110 is reading data from the S1D13708.
- WAIT# connects to RDY and is a signal output from the S1D13708 that indicates the SA-1110 must wait until data is ready (read cycle) or accepted (write cycle) on the host bus. Since SA-1110 accesses to the S1D13708 may occur asynchronously to the display update, it is possible that contention may occur in accessing the S1D13708 internal registers and/or display buffer. The WAIT# line resolves these contentions by forcing the host to wait until the resource arbitration is complete.
- The Bus Start (BS#) and RD/WR# signals are not used for this Host Bus Interface and should be tied high (connected to V_{DD}).
- The RESET# (active low) input of the S1D13708 may be connected to the system RESET.

4 StrongARM SA-1110 to S1D13708 Interface

4.1 Hardware Description

The SA-1110 microprocessor provides a variable latency I/O interface that can be used to support an external LCD controller. By using the Generic # 2 Host Bus Interface, no glue logic is required to interface the S1D13708 and the SA-1110.

A pull-up resistor is attached to WAIT# to speed up its rise time when terminating a cycle.

BS# (bus start) and RD/WR# are not used by the Generic #2 Host Bus Interface and must be tied high (connected to IO V_{DD}).

The following diagram shows a typical implementation of the SA-1110 to S1D13708 interface.

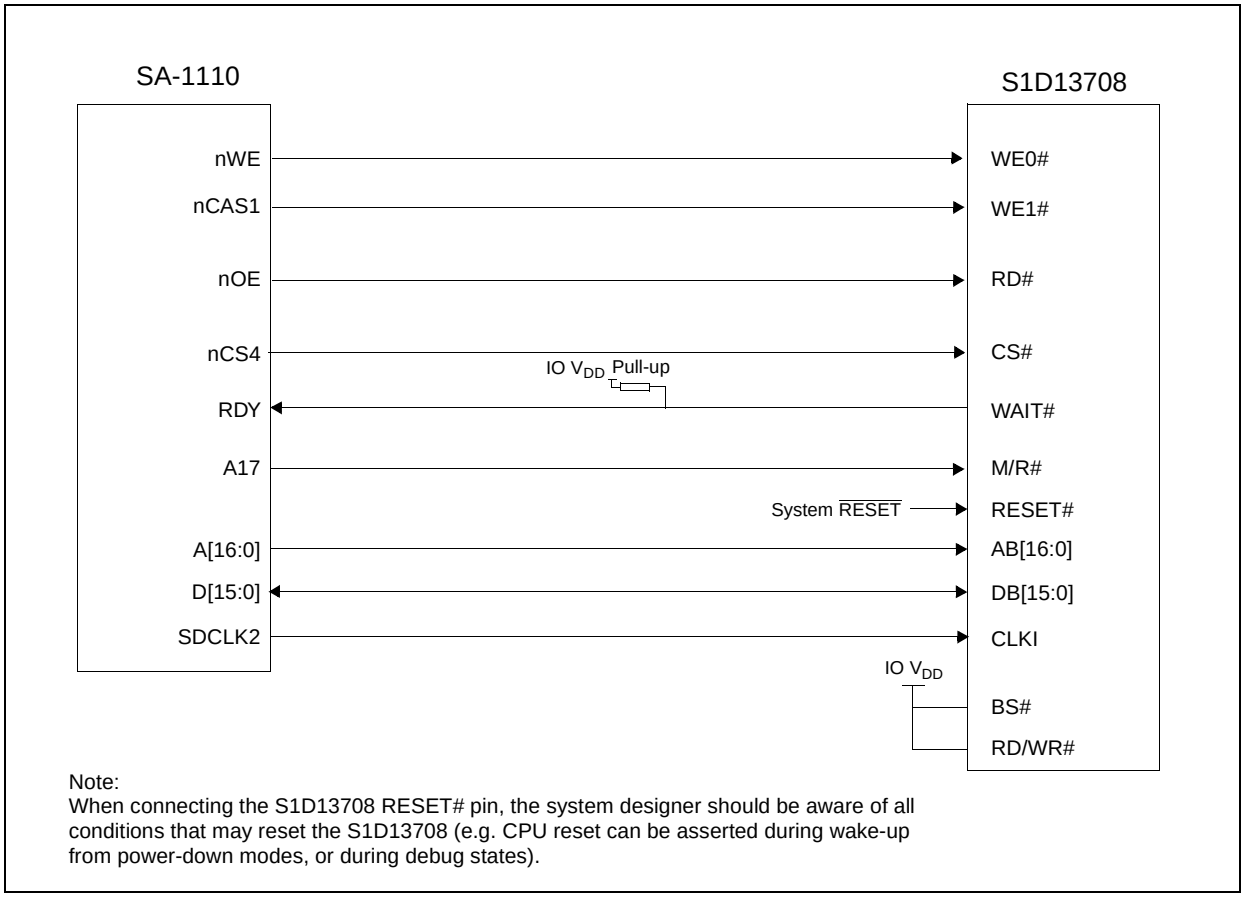


Figure 4-1: Typical Implementation of SA-1110 to S1D13708 Interface

4.2 S1D13708 Hardware Configuration

The S1D13708 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

The following table shows the configuration required for this implementation of a S1D13708 to SA-1110 interface.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13708 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[4, 2:0]	0100 = Generic #2 Little Endian Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	see Table "" for recommended setting	

= configuration for SA-1110

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for SA-1110

4.3 StrongARM SA-1110 Register Configuration

The SA-1110 requires configuration of several of its internal registers to interface to the S1D13708 Generic #2 Host Bus Interface.

- The Static Memory Control Registers (MSC[2:0]) are read/write registers containing control bits for configuring static memory or variable-latency IO devices. These registers correspond to chip select pairs nCS[5:4], nCS[3:2], and nCS[1:0] respectively. Each of the three registers contains two identical CNFG fields, one for each chip select within the pair. Since only nCS[5:3] controls variable-latency IO devices, MSC2 and MSC1 should be programmed based on the chip select used.

  Parameter RTx<1:0> should be set to 01b (selects variable-latency IO mode).

  Parameter RBWx should be set to 1 (selects 16-bit bus width).

  Parameter RDFx<4:0> should be set according to the maximum desired CPU frequency as indicated in the table below.

Table 4-3: RDFx Parameter Value versus CPU Maximum Frequency

CPU Frequency (MHz)	RDFx
57.3 - 85.9	1
88.5 - 143.2	2
147.5 - 200.5	3
206.4 - 221.2	4

  Parameter RDNx<4:0> should be set to 0 (minimum command precharge time).

  Parameter RRRx<2:0> should be set to 0 (minimum nCSx precharge time).

- The S1D13708 endian mode is set to little endian. To program the SA-1110 for little endian set bit 7 of the control register (register 1) to 0.
- The CLKI signal input to the S1D13708 from one of the SDCLK[2:1] pins is a derivative of the SA-1110 internal processor speed (either divide by 2 or 4). The S1D13708 Generic #2 Host Bus Interface has a maximum BCLK of 50MHz. Therefore, if the processor clock is higher than 100MHz, either divide the BCLK input using the S1D13708 configuration pins CNF[7:6] (see Table 4-2: “CLKI to BCLK Divide Selection”) or set SDCLK1/SDCLK2 to CPU clock divided by four using the DRAM Refresh Control Register (MDREFR bit 26 = 1 for SDCLK2, MDREFR bit 22 = 1 for SDCLK1).

4.4 Register/Memory Mapping

The S1D13708 is a memory-mapped device. The SA-1110 uses the memory assigned to a chip select (nCS4 in this example) to map the S1D13708 internal registers and display buffer. The S1D13708 uses two 128K byte blocks which are selected using A17 from the SA-1110 (A17 is connected to the S1D13708 M/R# pin).The internal registers occupy the first 128K bytes block and the 80K byte display buffer occupies the second 128K byte block.

Each variable-latency IO chip select is assigned 128M Bytes of address space. Therefore; if nCS4 is used the S1D13708 registers will be located at 4000 0000h and the display buffer will be located at 4002 0000h. These blocks are aliased over the entire 128M byte address space.

Note
If aliasing is not desirable, the upper addresses must be fully decoded.

5 Software

Test utilities and display drivers are available for the S1D13708. Full source code is available for both the test utilities and the drivers.

The test utilities are configurable for different panel types using a program called 13708CFG, or by directly modifying the source. The display drivers can be customized by the OEM for different panel types, resolutions and color depths only by modifying the source.

The S1D13708 test utilities and display drivers are available from your sales support contact or www.erd.epson.com.

6 References

6.1 Documents

- Intel Corporation, *StrongARM® SA-1110 Microprocessor Advanced Developer's Manual,* Order Number 278240-001.
- Epson Research and Development, Inc., *S1D13708 Hardware Functional Specification*, Document Number X39A-A-001-xx.
- Epson Research and Development, Inc., *S1D13708 Programming Notes and Examples*, Document Number X39A-G-003-xx.
- Epson Research and Development, Inc., *S5U13708B00C Rev. 1.0 ISA Bus Evaluation Board User Manual*, Document Number X39A-G-004-xx.

6.2 Document Sources

- Intel Developers Website: http://developer.intel.com.
- Intel Literature contact: 1(800) 548-4725.
- Epson Research and Development Website: http://www.erd.epson.com/.

7 Technical Support

7.1 EPSON LCD Controllers (S1D13708)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com/

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

7.2 Intel StrongARM SA-1110 Processor

INTEL

Intel Customer Support (ICS) for StrongARM: (800) 628-8686

Website for StrongARM Processor http://developer.intel.com/design/strong/

THIS PAGE LEFT BLANK

EPSON®

ENERGY SAVING EPSON

S1D13708 Embedded Memory LCD Controller

Connecting to a Micro-Controller via the Indirect Interface

Document Number: X39A-G-020-01

THIS PAGE LEFT BLANK

Table of Contents

THIS PAGE LEFT BLANK

List of Tables

List of Figures

THIS PAGE LEFT BLANK

1 Introduction

This application note describes the hardware and software routines required to interface the S1D13708 Embedded Memory LCD Controller to a micro-controller via the Indirect Interface. This document is not intended to cover all possible implementations, but provides a generic example of how such an interface can be accomplished.

The designs described in this document are presented only as examples of how such interfaces might be implemented. This application note is updated as appropriate. Please check the Epson Research and Development website at http://www.erd.epson.com for the latest revision of this document before beginning any development.

We appreciate your comments on our documentation. Please contact us via email at documentation@erd.epson.com.

2 Interfacing to a Micro-Controller

2.1 The Indirect Interface

Although the S1D13708 directly supports various CPU interfaces, the Indirect Interface allows micro-controllers that do not have a true CPU interface to use the S1D13708. In addition to being flexible and easy to implement on a micro-computer, this interface gives two other advantages.

- Reduced bus signal count, since no address lines are needed. There are two Indirect Interface modes, Mode 68 and Mode 80, both of which can handle 16-bit or 8-bit data transfers.
- S1D13708 asynchronous operation from the CPU.

Using Indirect Mode 68, 8-bit data transfer, only 12 IO signals are required, four for control and 8 for data. If 16-bit data transfers are required, an extra eight signals are needed for the expanded data bus and one more control signal. The S1D13708 can be set to run in asynchronous mode where the LCD controller runs independent of the CPU clock.

Indirect Mode 80 also allows 8-bit or 16-bit data transfers, but this mode requires one extra control signal when configured for 16-bit data mode.

3 S1D13708 Host Bus Interface

The S1D13708 directly supports many microprocessor busses, of which two are Indirect Interface modes, Mode 68 and Mode 80. Both modes can be used for micro-controllers where IO lines are available for use as a bus to the S1D13708.

Actual bus cycles will typically be controlled by software unless the micro-controller supports the Indirect Interface natively.

The Indirect Interface modes are selected by the S1D13708 based on the CNF[7:0] pin state on the rising edge of RESET#. After RESET# is released, the bus interface signals assume their selected configuration. For details on the S1D13708 configuration, please refer to Section 4.2, “S1D13708 Hardware Configuration” on page 14.

The S1D13708 clock (CLKI) is taken from the micro-controller bus clock in synchronous operation or from any other source for asynchronous bus operation. If the CLKI2 and XTAL inputs are not used they should be tied to ground, as with all other unused inputs.

For more information on the Indirect Interface mode interface, please refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

3.1 Indirect Mode Bus Interface Pin Mapping

The following table shows how the pins are connected in each of the Indirect modes.

Table 3-1: Mode 68 8-Bit Data Host Bus Interface Pin Mapping

S1D13708 Pin Names	Mode 68
AB[16:0]	connect to V_{SS}
DB[15:8]	connect to V_{SS}
DB[7:0]	D[7:0]
CS#	Chip Select
M/R#	A0
CLKI	BUSCLK
RD/WR#	R/W#
RD#	connect to V_{SS}
WE0#	EBL
WAIT#	not connected
RESET#	RESET#

Table 3-2: Mode 68 16-Bit Data Host Bus Interface Pin Mapping

S1D13708 Pin Names	connect to V_{SS}
AB[16:0]	connect to V_{SS}
DB[15:0]	D[15:0]
CS#	Chip Select
M/R#	A0
CLKI	BUSCLK
RD/WR#	R/W#
RD#	connect to V_{SS}
WE0#	EBL
WAIT#	not connected
RESET#	RESET#

Table 3-3: Mode 80 8-Bit Data Host Bus Interface Pin Mapping

S1D13708 Pin Names	connect to V_{SS}
AB[16:0]	connect to V_{SS}
DB[15:8]	connect to V_{SS}
DB[7:0]	D[7:0]
CS#	Chip Select
M/R#	A0
CLKI	BUSCLK
RD/WR#	RDL#
RD#	connect to V_{DD}
WE0#	WRL#
WAIT#	not connected
RESET#	RESET#

Table 3-4: Mode 80 16-Bit Data Host Bus Interface Pin Mapping

S1D13708 Pin Names	connect to V_{SS}
AB[16:0]	connect to V_{SS}
DB[15:0]	D[7:0]
CS#	Chip Select
M/R#	A0
CLKI	BUSCLK
RD/WR#	RDL#
RD#	RDU#
WE0#	WRL#
WAIT#	not connected
RESET#	RESET#

3.2 Host Bus Interface Signals

The S1D13708 Indirect Interface mode 68 and 80 host bus interface requires the following signals from a micro-controller.

- CLKI is a clock input which is required by the S1D13708 Host Bus Interface as a source for its internal bus and memory clocks. This clock is typically driven by the host CPU system clock. CLKI for the S1D13708 Indirect Interface can also be asynchronous with respect to the micro-controller bus clock, i.e. the clocks can be out of phase and/or different speeds. The only consideration to take into account is the software routines that control the bus cycles. The routines need to meet the bus interface timing requirements according to the S1D13708 bus clock.
- The data bus, DB[15:0] or DB[7:0], must be connected to bi-directional pins of the micro-controller lines.
- Chip Select (CS#) must be driven low when the S1D13708 is accessed by the micro-controller.
- M/R# [A0 (command signal)] selects between the signals on the data bus as being a register address or memory/register data.
- The RESET# (active low) input of the S1D13708 may be connected to the system RESET#.

The next three signals only apply to Mode 68:

- WE0# (EBL, active low) is the low byte enable input for both read and write cycles. This signal needs to be tied to a general purpose output line of the micro-controller.
- WE1# (EBH, active low) is the high byte enable input for both read and write cycles. This signal needs to be tied to a general purpose output line of the micro-controller for a 16-bit data bus, or to ground for an 8-bit data bus.
- RD/WR# (R/W#, active low) must be driven high for read accesses and low for write accesses. This signal needs to be tied to a general purpose output line of the micro-controller.

Note
Mode 68 requires the S1D13708 BS# and RD# signal inputs be tied to V_{SS}.

The next four signals only apply to Mode 80:

- RD/WR# (RDL#, active low) is the low byte enable for read cycles. This input signal needs to be tied to a general purpose output line of the micro-controller.
- RD# (RDU#, active low) is the high byte enable for read cycles. This input signal needs to be tied to a general purpose output line on the micro-controller for a 16-bit data bus, or to ground for an 8-bit data bus.
- WE0# (WRL#, active low) is the low byte enable for write cycles. This input signal needs to be tied to a general purpose output line of the micro-controller.
- WE1# (WRU#, active low) is the high byte enable for write cycles. This input signal needs to be tied to a general purpose output line of the micro-controller for a 16-bit data bus, or to ground for an 8-bit data bus.

Note
Mode 80 requires the S1D13708 BS# input signal be tied to V_{DD}.

4 Micro-Controller to S1D13708 Interface

4.1 Hardware Connections

The interface between the S1D13708 and a micro-controller requires no external logic if there are enough IO lines, and voltage levels are the same. As an example; the MicroChip P18C452 micro-controller has 34 IO lines available for use as general input/output lines, 1536 bytes of RAM, and runs 20Mhz at 3.3 volts.

To interface the S1D13708 to the P18C452 using Indirect Mode 68 requires only four output lines acting as control signals and eight bidirectional lines acting as the data bus. Using only 12 IO lines, a fully functioning S1D13708 LCD controller can be attached to the micro-controller. If a 16-bit data bus is desired, eight more bidirectional lines from the P18C452 can be connected to DB[15:8] and one more output line, acting as EBH, connected to the WE1# pin of the S1D13708.

The S1D13708 has 80K of embedded RAM which is used as the LCD display buffer. Depending on the LCD panel size, some of this memory can be used to store data for the micro-controller.

For example, using an Epson 160x160 ND-TFD panel in 8 bpp mode, the S1D13708 only requires 25K bytes of display buffer memory and will have 55K bytes of memory available for data storage which the P18C452 can use. If the bit-per-pixel setting is changed to 16 bpp, only 30K bytes of memory is available to the micro-controller for data storage.

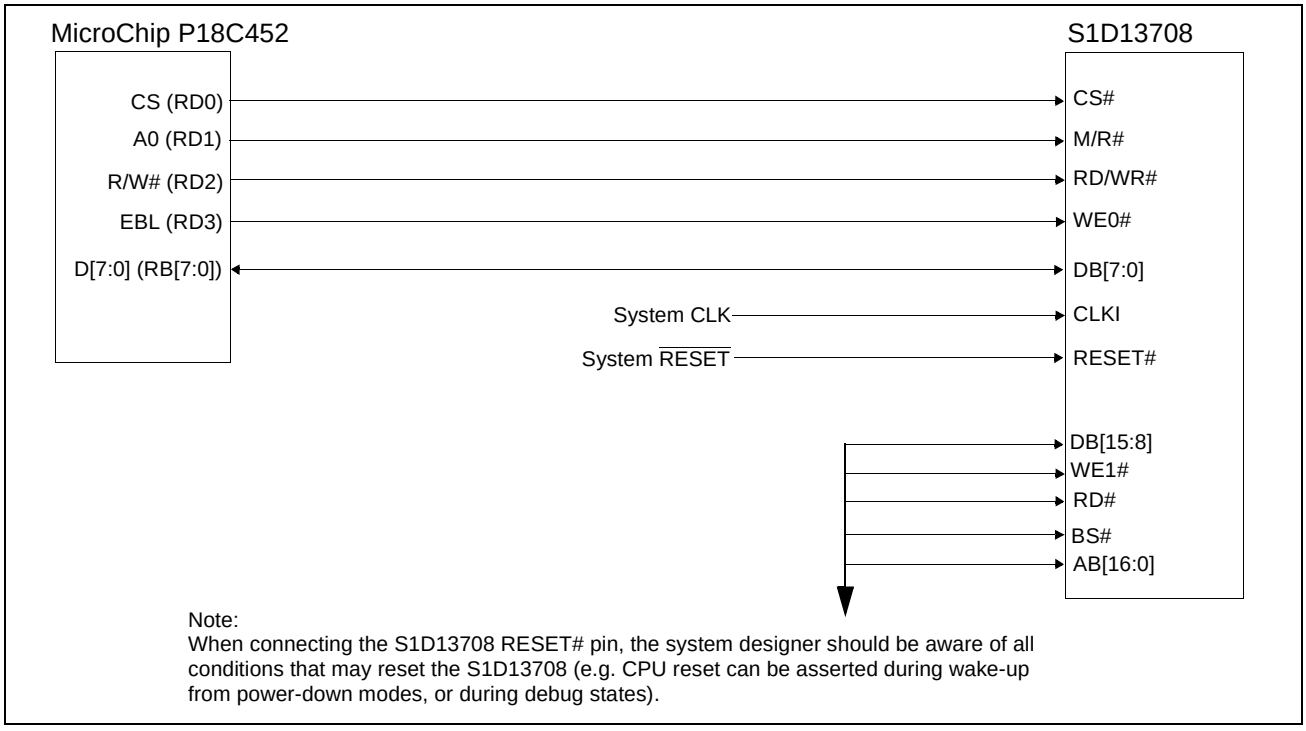


Figure 4-1: Typical Implementation of Micro-Controller to S1D13708 Interface

4.2 S1D13708 Hardware Configuration

The S1D13708 uses CNF7 through CNF0 to allow selection of the bus mode and other configuration data on the rising edge of RESET#. For details on configuration, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

The following table shows the configuration required for this implementation of the S1D13708 with the Indirect Interface. To select Mode 68, the BS# and RD# pins are tied to V_{SS}. To select Mode 80, only the BS# pin is tied to V_{DD}.

Table 4-1: Summary of Power-On/Reset Configuration Options

S1D13708 Pin Name	value on this pin at the rising edge of RESET# is used to configure: (1/0)	
	1	0
CNF[4, 2:0]	0111 = Indirect Mode Little Endian Host Bus Interface	
CNF3	GPIO pins as inputs at power on	GPIO pins as HR-TFT / D-TFT outputs
CNF5	Active high WAIT#	Active low WAIT#
CNF[7:6]	See Table 4-2: "CLKI to BCLK Divide Selection" for recommended setting	

= configuration for generic 8-bit processor

Table 4-2: CLKI to BCLK Divide Selection

CNF7	CNF6	CLKI to BCLK Divide
0	0	1:1
0	1	2:1
1	0	3:1
1	1	4:1

= recommended setting for Indirect Mode

4.3 Register/Memory Mapping

When the S1D13708 is in Indirect mode it is not a memory mapped device. It uses the protocol of first placing the register address on the data lines, then setting the register data on the data lines for the next cycle. The data lines are a multipurpose data bus, they can contain either the register address or the register data. The type of cycle, register address (command) or register data (data), is controlled by the A0 command signal.

The 80K byte display buffer memory of the S1D13708 is accessed through the use of four dedicated register sets. Registers REG[C0h], REG[C1h] and REG[C2h] contain the address of the memory pointer, and register REG[C4h] is the 'start' register. To access memory, REG[C2h], REG[C1h] and REG[C0h] are written with the desired memory address. Then, a command access to register REG[C3h] will start the memory access. Subsequent read/write data accesses will access the memory. The memory pointer address is automatically incremented after a memory data access.

The overhead for reading a large block of memory is the same as reading a small block of memory. Reading a single byte of memory will still require setting three address registers and a fourth access to the 'start' register.

For further information on the Indirect Interface bus cycles, please refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.

5 Software

To implement the bus cycles in software, refer to the *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx, for the bus timings. A minimum of three steps must be implemented in software to use the Indirect Interface. The steps for an 8-bit data transfer are:

- void command_write (char command_value)
- void data_write (char data_value)
- char data_read (void)

Example 1: The sequence for writing the S1D13708 register REG[7Ch] PIP+ Display Start with a value of 28h is:

- command_write(0x7C);
- data_write(0x28);

Example 2: The sequence for reading the S1D13708 register REG[26h] FPFRAME Pulse Start Register 0 is:

- command_write(0x26);
- value_returned= data_read();

Example 3: The sequence for writing to S1D13708 memory location 200h with two bytes of data is:

- command_write(0xC0);
- data_write(0x00);
- command_write(0xC1);
- data_write(0x02);
- command_write(0xC2);
- data_write(0x00);
- command_write(0xC3); // note that no ‘data’ value is needed for the ‘start’ register
- data_write(0xAA); // 1st byte of data written to 0x200
- data_write(0x55); // 2nd byte of data written 20 0x201

To read memory, the same sequence as above is used except a ‘data_read()’ is called. Note that the memory address is auto-incremented after a memory data access and the ‘start’ register REG[C3h] does not require a data value.

The following is a C implementation of the bus cycles:

```
// these are the simplified mode 68 bus cycles to mimic on the PIC general IO pins
#define CommandWriteStep1     0b10010100   // A0= low, RW=low
#define CommandWriteStep2     0b10011000   // CS=low, EBL= high
#define CommandWriteStep3     0b10010100   // EBL= low,CS=high
#define CommandWriteStep4     0b10110110   // a0= high, rw=high
#define DataWriteStep1        0b10010110   // rw=low
#define DataWriteStep2        0b10011010   // cs=low,ebl=high
#define DataWriteStep3        0b10010110   // cs=high,ebl=low
#define DataWriteStep4        0b10110110   // rw=high
#define DataReadStep1         0b10111010   // c78NORMAL & ~c78CS | c78EBL; CS=low, EBL= high

////////////////////////////////////////////////////////////////////////////
// routine to mimic a mode 68 command write cycle
//
void command_write (char command_value)
{
    // command write cycle
    PORTB= command_value;                 // place the register onto the data bus
    TRISB= 0;                             // make port as output
    LATD= CommandWriteStep1;
    LATD= CommandWriteStep2;
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    LATD= CommandWriteStep3;
    LATD= CommandWriteStep4;
    LATD= c78NORMAL;
    TRISB= 0xff;                          // set bus as Inputs
}

////////////////////////////////////////////////////////////////////////////
// routine to mimic a mode 68 data write cycle
//
void data_write (char data_value)
{
        PORTB= data_value; // place the register onto the data bus
    TRISB= 0;                             // make port as output
    LATD= DataWriteStep1;
    LATD= DataWriteStep2;
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    LATD= DataWriteStep3;
    LATD= DataWriteStep4;
    LATD= c78NORMAL;                      // back to original state
    TRISB= 0xff;                          // set bus as Inputs
}

////////////////////////////////////////////////////////////////////////////
// routine to mimic a mode 68 data read cycle
//
char return_data_read;
char data_read (void)
{
    TRISB= 0xff;                          // make port as output
    LATD= DataReadStep1;
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    return_data_read= PORTB;              // place the register onto the data bus
    LATD= c78NORMAL;                      // we can terminate cycle early for speed
    return return_data_read;
}
```

6 References

6.1 Documents

- Epson Research and Development, Inc., *S1D13708 Hardware Functional Specification*, document number X39A-A-001-xx.
- Epson Research and Development, Inc., *S1D13708 Programming Notes and Examples*, Document Number X39A-G-003-xx.

6.2 Document Sources

- Epson Research and Development Website: http://www.erd.epson.com/.

7 Technical Support

7.1 EPSON LCD Controllers (S1D13708)

Japan
Seiko Epson Corporation
Electronic Devices Marketing Division
421-8, Hino, Hino-shi
Tokyo 191-8501, Japan
Tel: 042-587-5812
Fax: 042-587-5564
http://www.epson.co.jp/

North America
Epson Electronics America, Inc.
150 River Oaks Parkway
San Jose, CA 95134, USA
Tel: (408) 922-0200
Fax: (408) 922-0238
http://www.eea.epson.com/

Taiwan
Epson Taiwan Technology
& Trading Ltd.
10F, No. 287
Nanking East Road
Sec. 3, Taipei, Taiwan
Tel: 02-2717-7360
Fax: 02-2712-9164
http://www.epson.com.tw/

Hong Kong
Epson Hong Kong Ltd.
20/F., Harbour Centre
25 Harbour Road
Wanchai, Hong Kong
Tel: 2585-4600
Fax: 2827-4346
http://www.epson.com.hk/

Europe
Epson Europe Electronics GmbH
Riesstrasse 15
80992 Munich, Germany
Tel: 089-14005-0
Fax: 089-14005-110
http://www.epson-electronics.de/

Singapore
Epson Singapore Pte., Ltd.
No. 1
Temasek Avenue #36-00
Millenia Tower
Singapore, 039192
Tel: 337-7911
Fax: 334-2716
http://www.epson.com.sg/

THIS PAGE LEFT BLANK